



Adilson de Jesus Semedo Gonçalves

Licenciado em Engenharia Eletrotécnica e das Telecomunicações

Controlo de Acesso ao Meio Fiável para Redes Ad Hoc

Dissertação para obtenção do Grau de Mestre em Engenharia Electrotécnica,
Sistemas e Computadores

Orientador : Doutor Rodolfo Oliveira, Professor Auxiliar, FCT-UNL

Júri:

Presidente: Prof. Doutor Luís Bernardo

Arguente: Prof. Doutor Pedro Medeiros

Vogais: Prof. Doutor Rodolfo Oliveira



**FACULDADE DE
CIÊNCIAS E TECNOLOGIA**
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2012

Controlo de Acesso ao Meio Fiável para Redes Ad Hoc

Copyright © Adilson de Jesus Semedo Gonçalves, Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Existe de certa forma um grupo de pessoas de que gostaria de expressar a minha mais sincera gratidão: Sem a vossa ajuda, seria muito difícil terminar este trabalho.

Em primeiro lugar, queria deixar em particular o meu mais profundo agradecimento ao meu orientador Prof. Doutor Rodolfo Oliveira, por toda sua preciosa ajuda e apoio prestado, bem como a orientação, motivação, pela exigência de método e rigor, pela incansável orientação científica, pela revisão cuidada e rigorosa na escrita e, ainda pelo seu profissionalismo e amizade demonstrada, o que considero determinante ao longo da elaboração desta dissertação, e que sem a sua ajuda não seria possível a sua implementação. *Muito Obrigado!*

À minha família e em especial à minha mãe, gostaria de expressar a minha mais profunda gratidão, por terem sido um suporte fulcral durante todos estes anos na minha caminhada académica. *Obrigado.*

Deixo ainda uma palavra de agradecimento e reconhecimento a todos os meus amigos, colegas, especialmente os alunos da sala do grupo de telecomunicações pela camaradagem e companhia, aos professores do departamento de engenharia electrotécnica e ainda ao meu grande amigo Jaquelino Rosa que considero como um irmão, que de alguma forma directa ou indirectamente contribuíram para a finalização desta dissertação. *Obrigado a todos!*

E por último, mas não menos importante à Nidia Santo, que tem sido uma boa companhia sobretudo nos momentos mais difíceis em que esteve sempre presente, pelo seu apoio e carinho. *Obrigado, Nidia!*

Sumário

O protocolo de controlo de acesso ao meio (MAC) com maior popularidade em redes locais sem fios e em redes ad hoc é o IEEE 802.11. Este protocolo foi desenhado essencialmente para transmissões do tipo *unicast*, apresentando elevada fiabilidade. Para transmissões do tipo *broadcast*, o protocolo não é fiável devido à elevada probabilidade de colisão provocada pelo uso da janela de contenção de valor fixo, e ainda pela falta de um mecanismo de Acknowledgement (ACK) das tramas *broadcast*. Infelizmente, muitos dos protocolos de encaminhamento geram elevadas quantidades do tráfego *broadcast* para troca de informações entre os nós, de forma a poderem descobrir e estabelecer rotas entre o nó de origem e o nó de destino. Este facto motiva o desenvolvimento de protocolos de controlo de acesso ao meio que apresentem elevada fiabilidade em cenários multi-hop e, simultaneamente, exibam elevado desempenho em termos do débito.

Esta dissertação propõe um novo protocolo MAC para redes multi-hop. O protocolo proposto efectua a optimização da janela de contenção adoptada pelos nós com base na estatística da ocupação do meio, com o objectivo de maximizar o débito do sistema numa rede multi-hop. Simultaneamente, o protocolo pretende obter valores de fiabilidade na transmissão das tramas que seja independentemente do tipo de tráfego (*unicast* ou *broadcast*) a suportar. Os resultados obtidos com a proposta descrita nesta dissertação exibem uma melhoria de desempenho, confirmando valores de fiabilidade constante para diferentes densidades de nós e quantidade de tráfego *unicast* e *broadcast*.

Palavras Chave: Controlo de Acesso ao Meio, Redes Ad Hoc, Redes Sem Fios.

Abstract

Nowadays the standard IEEE 802.11 is the most popular protocol for wireless local area networks and ad hoc networks. The medium access control scheme proposed in IEEE 802.11 was designed primarily for unicast transmissions, which exhibit high reliability. For broadcast transmissions the MAC scheme proposed in IEEE 802.11 is unreliable due to the absence of an acknowledgement scheme, which origins a high rate of collisions. Unfortunately, most of the routing protocols proposed for ad hoc networks generate large amounts of broadcast traffic, namely to exchange information related to the network topology, so that they can discover and establish new routes between source and destination nodes. This fact motivates the development of new medium access control schemes that should exhibit both high probability of successful transmission and high throughput in multi-hop scenarios for both types of broadcast and unicast traffic.

This work proposes a new contention-based MAC protocol for multi-hop networks. The proposed protocol optimizes the individual contention window based on statistics collected from the channel. Simultaneously, the protocol seeks to obtain a constant probability of collision, independently of the types of traffic (broadcast/unicast). The results achieved with the proposed protocol shown an improvement in terms of throughput, and confirm that the probability of successful transmission is constant and does not depend on traffic's type nor on the number of nodes.

Keywords: Medium Access Control, Ad Hoc Networks, Wireless Networks.

Conteúdo

Agradecimentos	iii
Sumário	v
Abstract	vii
Acrónimos	xv
1 INTRODUÇÃO	1
1.1 Motivação	1
1.2 Objectivos e Contribuições	4
1.3 Estrutura da Dissertação	5
2 TRABALHO RELACIONADO	7
2.1 Introdução	7
2.2 Controlo de Acesso ao Meio da norma IEEE 802.11	7
2.3 Broadcast Medium Window (BMW)	11
2.4 BMMM e LAMM	14
2.5 Esquema de <i>broadcast</i> duplicado	15
2.6 LBP, DBP e PBP	19
2.7 Dual Busy Tones	21
2.8 MyopicMAC	24
2.9 Resumo	27
2.10 Problemas associados às redes multi-hop	27
3 MODELO TEÓRICO	29
3.1 Introdução	29
3.2 Probabilidade de Acesso Individual	29
3.3 Cenário Multi-hop	31
3.4 Débito	34
3.5 Validação do Modelo	35
3.5.1 Notas finais	35

4	PROTOCOLO MAC	37
4.1	Topologia	37
4.2	Comparação dos Protocolos Myopic e 802.11	39
4.3	Análise dos Parâmetros de Optimização	40
4.4	Implementação Prática	44
4.4.1	Probabilidade de sucesso	44
4.4.2	Probabilidade de <i>idle slots</i>	44
4.5	Optimização da Janela de Contenção	45
5	ANÁLISE DO DESEMPENHO	49
5.1	O simulador	49
5.2	Parâmetros Usados no Simulador	50
5.3	Aquisição de Dados e Cálculo do Débito	51
5.4	Desempenho	52
6	CONCLUSÕES	55
6.1	Conclusões	55
A	Anexo A	57
B	Anexo B	59
	Bibliografia	62

Lista de Figuras

1.1	Modelo de comunicação em redes móveis estruturadas.	2
1.2	Rede móvel móvel ad hoc. Os círculos denotam o alcance de transmissão dos nós, onde a comunicação entre o nó A e o nó D utiliza as ligações formadas pelos nós B e C.	3
2.1	Mecanismo de acesso Básico.	10
2.2	Mecanismo de acesso empregando RTS/CTS.	11
2.3	O nó 5 efectua <i>broadcast</i> da trama aos nós vizinhos.	14
2.4	Ilustração de esquema BACK.	17
2.5	Modelo de enfileiramento dos pacotes de dados e broadcast na camada de rede e fila de prioridade na camada MAC.	18
2.6	Diagrama de estados do emissor (tirada do [1]).	22
2.7	Diagrama de estados do receptor (tirada do [1]).	23
2.8	Controlador PI $C(\tilde{\tau})$ (tirada do [2]).	26
2.9	Terminal (nó) escondido.	27
2.10	Terminal (nó) exposto.	28
3.1	Ilustração de uma cadeia de Markov.	30
3.2	Cenário <i>single hop</i>	32
3.3	Número médio dos nós no raio de transmissão do nó A é 5 (nós A, B, D, E,F).	33
3.4	Área dos nós contabilizados $n_1 - 2$ (C, G e H), para expoente da parcela $(1 - \tau_2)$ na expressão (3.11).	34
3.5	Resultados numéricos do modelo	36
4.1	T-Alcance de transmissão (100m) e S-Alcance de interferência (111.4m). . .	37
4.2	Topologia dos nós em grelha distanciados de 100m.	38
4.3	Ilustração dos nós vizinhos na grelha.	38
4.4	Relação de débito dos protocolos MyopicMAC e IEEE 802.11 com a variação do Cw_{min}	39
4.5	Ponto de Cw^* onde se maximiza o débito.	40
4.6	P_i óptimo (P_i^*).	41

4.7	$P_{sucesso}$ óptimo ($P_{sucesso}^*$).	42
4.8	Cw^* para diferente número dos nós (n).	42
4.9	Ilustração de probabilidade <i>idle slots</i> (P_i^*) e de sucesso ($P_{sucesso}^*$) óptimos com a variação dos nós (n).	44
4.10	Tamanho do <i>buffer</i> de $\Delta t = t2 - t1$ <i>time slots</i>	46
4.11	Variação do \bar{P}_{idle} com o tempo.	46
4.12	Desvio padrão do P_i	47
4.13	Diagrama de blocos do algoritmo.	47
5.1	Arquitectura ns2 em que o OTcl representa a interface do utilizador, C++ o mecanismo interno e o Tclcl interliga C++ ao OTcl.	50
5.2	Débito dos protocolos IEEE 802.11 e do protocolo proposto no Capítulo 4 em função da percentagem de tramas <i>broadcast</i> geradas.	53
5.3	Probabilidade de sucesso dos protocolos IEEE 802.11 e do protocolo pro- posto em função da percentagem tramas <i>broadcast</i> geradas.	54

Lista de Tabelas

2.1	<i>Time Slot</i> , valores mínimos e máximos da janela de contenção de acordo com tecnologias adoptada no nível físico especificado pela norma 802.11. . . .	9
3.1	Parâmetros utilizado para a validação do modelo.	35
4.1	Variação do P_i^* com o número dos nós (n).	43
5.1	Parâmetros utilizado na simulação.	50

Acrónimos

ACK Trama de Acknowledge

BEB Binary Exponential Backoff

BT Backoff Time

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance

CTS Clear To Send

CW Contention Window

DCF Distributed Coordination Function

DIFS Distributed InterFrame Space

DSSS Direct-Sequence Spread Spectrum

EIFS Extended InterFrame Space

FHSS Frequency Hoping Spread Spectrum

IEEE Institute of Electrical and Electronics Engineers

IP Internet Protocol

IR Infra Red

LAN Local Area Network

MAC Medium Access Control

MANET Mobile Ad-hoc NETwork

NAV Network Allocator Vector

Nó Aparelho capaz de comunicar com outros aparelhos numa rede. Exemplos de nós podem ser, telefones celulares, *laptops* ou PDAs etc.

Ns Network Simulator

OSI Open System Interconnection

PCF Point Coordination Function

PDA Personal Digital Assistants

PHY PHYsical Layer

RTS Resquest To Send

SIFS Short InterFrame Space

WLAN Wireless Local Area Network

Capítulo 1

INTRODUÇÃO

1.1 Motivação

A evolução ocorrida no campo das comunicações sem fios contribuirão, num futuro próximo, para que a informação e os recursos possam ser acedidos e utilizados em qualquer lugar e em qualquer momento. Dado o grande crescimento do segmento de computadores pessoais portáteis e PDAs (*Personal Digital Assistants*), estima-se que em alguns anos, dezenas de milhares de pessoas terão um *laptop*, *palmtop* ou algum género de PDA. Independente do tipo de dispositivo portátil, a maior parte desses equipamentos deverão estar equipados com *transceivers*¹ de modo a comunicarem com outros dispositivos. A este processo dá-se o nome de computação móvel, pelo qual o dispositivo não precisará de uma posição fixa para estabelecer comunicação com outros dispositivos. Este paradigma permite que os utilizadores desse ambiente tenham acesso aos serviços em qualquer local e, acima de tudo, permitindo-lhes o acesso às diversas redes de comunicações quando se deslocam.

As comunicações móveis subdividem-se em dois grandes tipos, sendo uma infra-estruturada e a outra não infra-estruturada. No tipo infra-estruturada existe uma entidade central de controlo (ver Figura 1.1) e no tipo não infra-estruturada o controlo da rede é realizada de forma distribuída por todos os nós que constituem a rede (ver Figura 1.2).

As redes ad hoc móveis são um exemplo de rede não infra-estruturada, tal como é ilustrado na Figura 1.2. Estas, são constituídas por um conjunto de terminais equipados com *transceivers* sem fios que podem comunicar com outros terminais na ausência de qualquer

¹dispositivo para transmitir e receber no mesmo equipamento

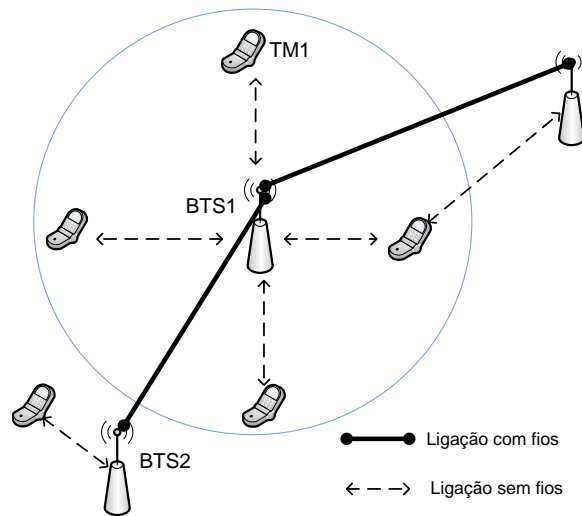


Figura 1.1: Modelo de comunicação em redes móveis estruturadas.

infra-estrutura fixa. A ausência da infra-estrutura, como por exemplo as várias estações base ou antenas fixas de sistemas celulares, faz com que as redes ad hoc sejam uma forma radicalmente diferente de outras redes locais (LANs - *Local Area Network*) sem fios. A comunicação de um terminal móvel (nó) numa rede infra-estruturada, tal como sucede nas redes celulares, é sempre estabelecida com a estação base (BTS - *Base transceiver station*) que é fixa. Um nó de uma rede ad hoc pode comunicar directamente com outros nós se estiveram dentro do seu raio de alcance. Para comunicar com um nó que não esteja dentro do alcance rádio, são usados nós intermédios para encaminhar os pacotes de dados até ao nó de destino, ou seja, os nós têm o papel de armazenar e encaminhar pacotes como um router (encaminhador). Como não necessitam de estação base, as redes ad hoc podem ser estabelecidas rapidamente sem ter de efectuar um planeamento avançado ou o estabelecimento de uma infra-estrutura cara, sendo estas redes ideais para aplicações onde não se pode, ou não faz nenhum sentido, instalar uma rede fixa. Tipicamente as aplicações deste tipo de redes incluem redes de comunicações militares no campo de batalha, redes de operação de busca e resgate, comunicações em operações submarinas, monitorização do meio ambiente e exploração espacial. Devido à sua facilidade de implementação e ao custo relativamente baixo, as redes ad hoc também são usadas noutros cenários que incluem redes de *laptops* em salas de conferência, redes de equipamentos electrónicos noutros cenários (por exemplo, televisão, computador, impressora, entre outros) para formar uma

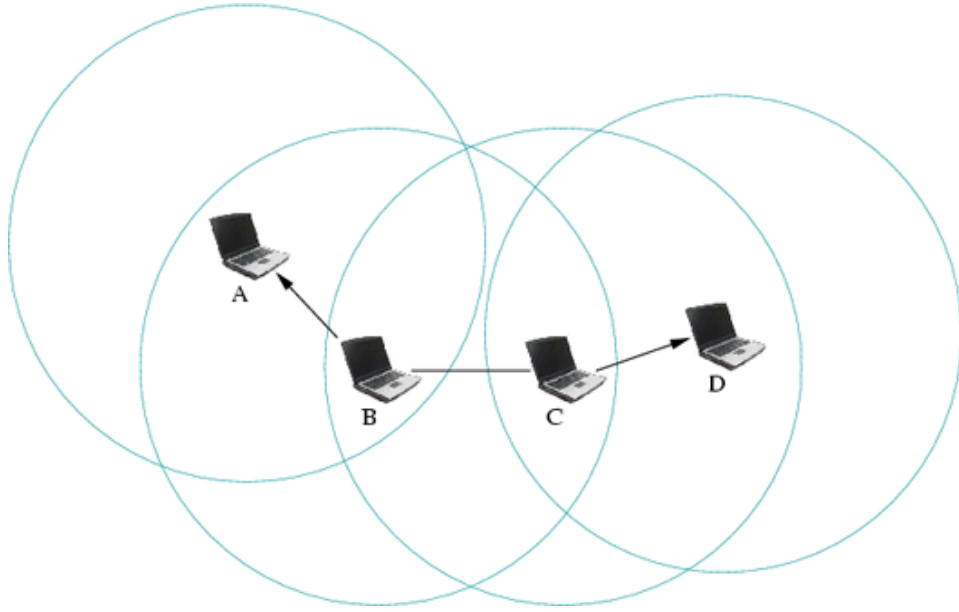


Figura 1.2: Rede móvel móvel ad hoc. Os círculos denotam o alcance de transmissão dos nós, onde a comunicação entre o nó A e o nó D utiliza as ligações formadas pelos nós B e C.

rede de área local, redes de robôs móveis e brinquedos sem fios [3], [4]. Recentemente, tem-se verificado um crescente interesse no uso de redes de sensores sem fios ad hoc para realizar a vigilância e operações de rastreamento [5].

Contudo, o desenho das redes ad hoc enfrenta vários desafios, originados pelo facto dos nós (origem, destino e encaminhadores) poderem ser móveis. À medida que o alcance de transmissão sem fios é limitada (e numa rede ad hoc é fundamental determinar e manter as rotas das ligações), a ligação entre um par dos nós pode quebrar devido a um deles se mover para fora do alcance rádio do outro, causando assim mudanças na topologia da rede, o que torna a qualidade de ligação das redes sem fios imprevisível. Outro grande desafio é a ausência de uma entidade de controlo centralizado. Todas as funções de rede, como por exemplo, determinar a topologia de rede, múltiplo acesso e encaminhamento (*routing*) de pacotes sobre os caminhos (*path*)² de múltiplos saltos (*Multi-Hop*), devem ser realizadas de forma distribuída. Estes desafios são extremamente ambiciosos em virtude das baterias dos dispositivos móveis terem um tempo de vida relativamente baixo e uma largura de banda limitada.

²Série de ligações que liga um nó origem a um nó de destino

Os desafios exibidos pelas redes ad hoc devem ser abordados em todos os níveis das camadas da pilha de protocolos. A camada física deve lidar com o *pathloss*³ (perdas de propagação), *fading*⁴ (desvanecimento), e interferência entre utilizadores para manter a comunicação estável entre os pares dos nós. A camada de ligação de dados (DLL - *Data Link Layer*), na qual possui a sub-camada MAC, deve efectuar uma ligação com alta fiabilidade e resolver a questão de acesso dos utilizadores de forma assíncrona e distribuída. Da mesma forma as camadas de rede, transporte e aplicação ficam encarregadas de lidar com os protocolos de encaminhamento de pacotes, atrasos/perdas dos pacotes e suporte das desconexões, respectivamente. É sobre estes protocolos que assenta esta dissertação, em particular o protocolo da sub-camada MAC (*Medium Access Control*), que será abordado nos próximos capítulos.

Embora esta área tenha recebido muita atenção nos últimos anos, a ideia de redes ad hoc surgiu em 1970, quando o DARPA (*Defense Advanced Research Projects Agency*) nos Estados Unidos de América (EUA), patrocinou o projecto PRNET (*Packet Radio Network*) [6]. E assim foi seguido pelo projecto SURAN (*Survivable Adaptive Radio Network*) na década de 1980 [7]. Contudo, o interesse nesta área cresceu com rapidez na década de 90 devido à popularidade dos aparelhos digitais móveis tais como *laptops* e PDAs. Com o aumento da popularidade da Internet, o interesse em desenvolver os protocolos de comunicação para as redes móveis ad hoc tem progredido. O interesse no desenvolvimento de protocolos para redes ad hoc baseados em IP (*Internet Protocol*) é hoje um facto consolidado, tendo sido formado um grupo de trabalho para as redes móveis ad hoc (MANET - *Mobile Ad Hoc Networking*) no IETF (*Internet Engineering Task Force*) [8].

1.2 Objectivos e Contribuições

O objectivo desta dissertação consiste em melhorar o protocolo descrito em [2], que só se aplica a redes do tipo *single-hop*, onde dois nós comunicam directamente entre si sem a necessidade de algoritmos de encaminhamento. O trabalho desenvolvido nesta dissertação

³Enfraquecimento do sinal devido à distância percorrida.

⁴Soma dos vários sinais no receptor, tornando-o imperceptível.

pretende melhorar o débito e a probabilidade de sucesso das transmissões em redes *multi-hop* sem fios e em cenários de transmissões *broadcast*⁵.

As contribuições deste trabalho incluem:

- O levantamento bibliográfico dos protocolos MAC que pretende aumentar a fiabilidade da transmissão em ambientes multi-hop (Capítulo 2);
- A elaboração e validação de um modelo simplificado do desempenho de um protocolo MAC baseado em contenção (Capítulo 3);
- A exploração do modelo de desempenho do protocolo MAC para especificar critérios de otimização no qual se poderão basear eventuais protocolos (Capítulo 4);
- O desenvolvimento e a caracterização de desempenho de um protocolo MAC multi-hop baseados nos critérios de otimização encontrados (Capítulo 5).

1.3 Estrutura da Dissertação

A dissertação encontra-se organizada em seis capítulos, conforme se resume em seguida. No Capítulo 2 “Trabalho Relacionado”, é apresentada uma breve introdução dos desafios levantadas pelas redes ad hoc, descrevendo alguns protocolos já propostos para essas redes sem fios. Também são apresentadas as suas características bem como a identificação dos pontos positivos e negativos de cada um, em termos da fiabilidade e do débito.

Em relação ao Capítulo 3 “Modelo Teórico”, é apresentado um modelo que permite uma análise da transmissão de tramas a 2 *hops*, sendo validado em termos do débito e da probabilidade de sucesso de transmissão.

No Capítulo 4 “Protocolo MAC Multi-Hop”, o modelo apresentado no capítulo 3 é explorado de forma a encontrar alguns princípios de otimização que possam ser explorados por protocolos MAC. Foi realizada uma análise numérica que valida a maximização do débito, sendo proposto um protocolo capaz de implementar esse princípio de otimização. A parte final deste capítulo descreve o modo de funcionamento do protocolo, bem como a sua implementação prática.

⁵O nó envia uma trama para todos os vizinhos a 1 *hop*

No Capítulo 5 “Análise do desempenho”, é realizada uma breve introdução ao simulador ns-2 [9] utilizado para aferir o desempenho do protocolo proposto, analisando-se os resultados obtidos.

Finalmente no Capítulo 6 “Conclusões”, resumem-se as principais conclusões obtidas neste trabalho.

Capítulo 2

TRABALHO RELACIONADO

2.1 Introdução

A sub-camada MAC (*Medium Access Control*) das redes sem fios distribuídas apresenta desafios interessantes, dadas as dificuldades causadas pelos erros de transmissão, colisão entre tramas e os efeitos inerentes às redes multi-hop (secção 2.10). Estas dificuldades são ainda mais severas quando os protocolos suportam comunicações do tipo *Multicast* ou *broadcast*. Esse suporte é necessário para oferecer uma qualidade de serviço aceitável em diversos cenários de aplicação. Dessa forma, é necessário ter mecanismos de MAC eficientes, os quais deverão maximizar o débito e apresentar elevada justiça de acesso entre os nós.

Nos últimos anos têm-se publicado muitos trabalhos abordando o estudo dos protocolos da camada MAC, com o objectivo de atingir maior fiabilidade de comunicação e elevado débito. Neste capítulo, analisam-se alguns dos principais trabalhos que propõem melhorar o desempenho do protocolo da camada MAC.

2.2 Controlo de Acesso ao Meio da norma IEEE 802.11

No protocolo IEEE 802.11, o mecanismo fundamental para o acesso ao meio é denominado de DCF (*Distributed Coordination Function*). Este é um esquema de acesso aleatório, não tendo uma entidade central para coordenar a transmissão das tramas. A coordenação

é distribuída e baseada no mecanismo de janela de contenção do protocolo CSMA/CA¹ (*Carrier Sense Multiple Access/Collision Avoidance*). É usado o mecanismo de acesso denominado BEB (*Binary Exponential Backoff*) para diminuir as colisões entre tramas retransmitidas. A norma também define uma forma de acesso que é opcional, PCF (*Point Coordination Function*), onde a gestão do acesso é centralizada num único nó.

Como neste trabalho são abordados esquemas de controlo distribuído, é somente descrito o esquema de funcionamento DCF. O esquema DCF descreve duas técnicas para o envio das tramas. O esquema por omissão é definido como *two-way handshaking* denominado de mecanismo **acesso básico**. Este mecanismo é caracterizado pela transmissão imediata de um Acknowledgement (ACK), assim que o nó receptor recebe uma trama com sucesso.

Para além do acesso básico, é ainda proposto uma técnica opcional denominada de *four way handshaking*, conhecido como mecanismo ***Request-To-Send*** (RTS)/***Clear-To-Send*** (CTS). Antes de transmitir tramas de dados, um nó emissor e receptor trocam tramas RTS/CTS. Quando o nó emissor deseja enviar tramas de dados, primeiro envia uma trama do tipo RTS. O nó de destino faz o Acknowledgment da recepção da trama RTS com uma trama CTS. Depois, o emissor envia a trama de dados e o receptor responde com outra trama do tipo ACK para confirmar a correcta recepção dos dados. Como a colisão ocorre somente nas tramas RTS, e isto é detectado pela falha (não recepção) das tramas CTS, o mecanismo de RTS/CTS permite aumentar o desempenho do sistema ao diminuir a duração da colisão quando são transmitidas mensagens de grande tamanho.

Funcionamento DCF

Nesta sub-secção é apresentada de uma forma sumariada o funcionamento da *Distributed Coordination Function* (DCF), especificado no protocolo IEEE 802.11. Em [10] encontra-se informação mais detalhada acerca deste modo de funcionamento.

Quando um nó pretende transmitir tramas de dados, primeiro verifica o canal, se estiver livre durante o intervalo de tempo igual a DIFS (*Distributed Interframe Space*), o nó transmite. Pelo contrário, se o canal estiver ocupado (imediatamente ou durante o período DIFS), o nó persiste em monitorizar o canal até que este se encontre novamente

¹Inspirado no CSMA das redes fixas, com o princípio de: "*escuta antes de transmitir*"

livre após um período DIFS. Nesta etapa, o nó gera um período de contenção (BT-*Backoff Time*) aleatório antes de transmitir a trama para minimizar a probabilidade de colisão com tramas enviadas por outros nós.

O método DCF adopta um esquema BEB (*Binary Exponential Backoff*) que consiste em gerar, de acordo com uma distribuição probabilística uniforme, um período de contenção de cada vez em que é necessário aceder ao meio. O período de contenção é obtido da seguinte forma

$$BT = \text{uniform}(0, Cw - 1) \times \text{SlotTime} \quad (2.1)$$

onde o valor de Cw é a janela de contenção, e depende do número de transmissões falhadas de uma dada trama. Na primeira tentativa de transmissão, Cw é igual ao valor mínimo da janela de contenção Cw_{min} . Depois de cada transmissão sem sucesso, Cw é duplicado, até ao valor máximo $Cw_{max} = 2^m Cw_{min}$, em que o m representa o número de tentativas sem sucesso. Os valores Cw_{min} e Cw_{max} explicitos na versão inicial da norma IEEE 802.11 encontram-se especificados na tabela 2.1 para diferentes tecnologias adoptas no nível físico PHY.

PHY	Time Slot(δ)	Cw_{min}	Cw_{max}
FHSS	50 μs	16	1024
DSSS	20 μs	32	1024
IR	8 μs	64	1024

Tabela 2.1: *Time Slot*, valores mínimos e máximos da janela de contenção de acordo com tecnologias adoptada no nível físico especificado pela norma 802.11.

O contador do período de contenção é decrementado de acordo com a expressão (2.2)

$$BT_{novo} = BT_{actual} - 1 \quad (2.2)$$

sempre que o canal esteja livre. Quando é detectada uma ou mais transmissões no canal, a subtracção de BT_{actual} em (2.2) não se verifica, e a subtracção é reactivada assim que o canal estiver novamente livre após um período de tempo igual a DIFS. Um nó transmite sempre que o BT_{novo} atingir o valor 0.

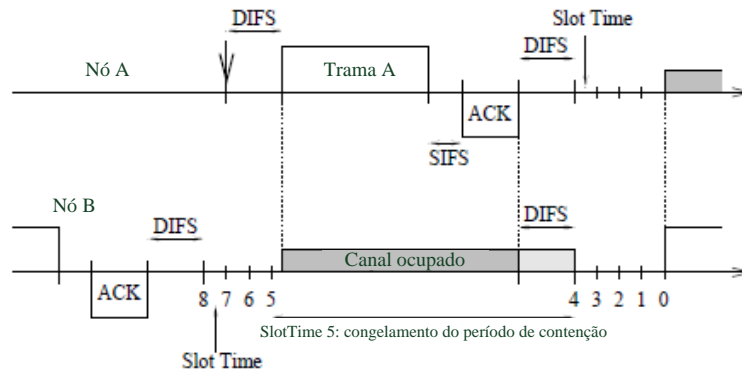


Figura 2.1: Mecanismo de acesso Básico.

A Figura 2.1, ilustra o mecanismo de acesso básico, onde dois nós A e B partilham o mesmo canal. No final da transmissão da trama, o nó B espera um intervalo DIFS e, de seguida, escolhe um período de contensão de acordo com a expressão (2.1), ou seja, igual a 8 antes de transmitir a sua próxima trama. Presume-se que a primeira trama do nó A chega no tempo indicado pela seta na Figura 2.1. Depois de terminar o período DIFS, a trama é transmitida. Note-se que a transmissão da trama A ocorre quando o valor de contensão do nó B, é igual a 5. Devido ao canal estar ocupado, o período de contensão do nó B é congelado no valor 5, e a decrementar do contador de contensão é reactivado após o canal estar livre durante um período DIFS.

A trama de acknowledgement (ACK) é transmitida pelo nó de destino para sinalizar uma recepção com sucesso. O ACK é imediatamente transmitido no fim da recepção da trama, passado um período de tempo chamado SIFS (*Short InterFrame Space*). Como o $SIFS < DIFS$, nenhum dos outros nós são capazes de detectar o canal ocupado durante um período de DIFS até ao fim do ACK. Se o nó emissor não receber o ACK dentro de um intervalo de tempo $ACK_Timeout$ especificado pela norma, o mesmo considera-se que a transmissão da trama não teve sucesso (devido a uma colisão ou erro no meio). Pelo que, será reagendada uma nova transmissão da trama com a janela de contensão aumentada para o dobro.

No mecanismo RTS/CTS (ver Figura 2.2), o nó emissor e receptor antes de enviarem as tramas de dados para o canal, trocam pequenas tramas RTS/CTS, que incluem

informações sobre o tamanho da trama que será transmitida. Esta informação pode ser lida por todos os nós que escutam o meio, que assim irão actualizar o NAV (*Network Allocation Vector*) contendo a informação do período em que o canal ficará ocupado. Assim, quando um nó estiver *escondido* do emissor ou do receptor, ao detectar uma das tramas RTS/CTS, irá atrasar a sua transmissão, e assim evita a colisão.

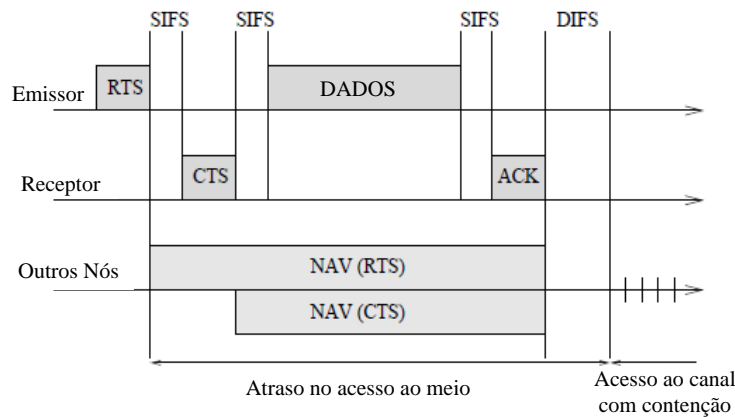


Figura 2.2: Mecanismo de acesso empregando RTS/CTS.

2.3 Broadcast Medium Window (BMW)

O protocolo estudado em [11] tem como principal preocupação a transmissão fiável das tramas aos seus vizinhos. Este protocolo trata cada pedido de transmissão *broadcast* como múltiplos pedidos *unicast*. Cada trama *unicast* é processada usando o protocolo MAC IEEE 802.11 DCF (CSMA/RTS/CTS/DATA/ACK). Cada nó armazena 3 listas:

1. *Neighbor List* - mantém a lista dos nós vizinhos;
2. *Send Buffer* - contém a lista das tramas que já foram transmitidas;
3. *Receiver Buffer*- lista do número de sequência recebida.

Um nó, ao receber uma trama, actualiza logo a sua tabela *Neighbor List* que contém a lista dos nós vizinhos. Cada nó armazena as tramas que transmite pelo meio numa tabela *Send Buffer*, pois poderá ser necessário a sua retransmissão, e só as apaga do *Buffer* quando todos os receptores as recebem. Finalmente cada nó mantém na tabela *Receiver Buffer* o número de sequência recebido correspondente a uma determinada trama. O nó

emissor, ao transmitir uma trama do tipo RTS para um nó de destino especifica sempre o seu número de sequência. Assim, quando o receptor recebe uma determinada trama de dados, examina a tabela *Receiver Buffer* para saber se falta algum número de sequência. Se assim for, o receptor notificará o emissor dessa perda de número de sequência na trama do tipo CTS.

Quando um nó tiver tramas para transmitir, primeiro faz o *sensing* do canal e entra na fase de *collision avoidance* (CSMA/CA) semelhante ao 802.11. Após completar a fase de *collision avoidance* e o canal está livre, o nó envia RTS para um dos seus vizinhos, especificando na trama a gama do número de sequência que foi enviado inicialmente e o actual (inicial - corrente). O receptor ao receber a trama RTS, examina a sua tabela *Receiver Buffer* e verifica o número de sequência das tramas que ainda precisa. Se estiver a faltar alguma dentro daquela gama do número de sequência, o nó responde com outra trama CTS onde notifica o sucedido. Contudo, se só faltar o número de sequência corrente, a trama CTS notifica o emissor de igual modo. Todos os outros nós no meio ao ouvirem a trama RTS irão se conter o tempo suficiente de maneira a não interferirem na transmissão da trama CTS/DATA/ACK que será transmitida. Depois de receber o CTS, o nó emissor transmite os dados correspondentes ao número de sequência especificado na trama CTS. Os restantes nós ao ouvirem a trama CTS irão se conter um intervalo de tempo suficiente para não se interferirem na transmissão de DATA/ACK. Ao receber os dados, o nó de destino actualiza a sua tabela *Receiver Buffer* e responde com um ACK. Por fim, o nó emissor coloca os dados no *send buffer* e escolhe o próximo vizinho na tabela *Neighbor List*, repetindo todo o procedimento anterior. No primeiro instante os nós não têm conhecimento sobre os seus vizinhos. A transmissão *broadcast* não será fiável até os vizinhos serem detectados.

Na Figura 2.3 apresenta-se uma pequena ilustração do protocolo *BMW* onde o nó 5 efectua *broadcast* para os nós 1, 2, 3 e 4 que estão dentro do seu raio de alcance. Inicialmente, o nó 5 terá de descobrir os vizinhos. Ao encontrar o nó 1, o nó 5 envia a trama RTS com uma gama de números de sequência de 0 – 0, onde o primeiro 0 significa sequência inicial e o segundo a sequência corrente, pois como ainda não houve nenhuma

transmissão de dados, tanto a sequência inicial como a actual tem o valor 0. O nó 1 ao receber a trama RTS, responde com a trama CTS com a sequência 0. Como já foi mencionado acima, os restantes nós contêm-se durante um intervalo de tempo suficiente até que os nós 5 e 1 terminem de trocar as tramas. O nó 5, ao receber a trama CTS com a sequência 0 transmite a trama correspondente. Assim que o nó 1 recebe os dados, actualiza a sua tabela *Receiver Buffer* com a indicação de que já tem a trama de dados da sequência 0. Por exemplo, supõe-se que os nós 1, 2 e 4 receberam correctamente as tramas de dados da sequência 0, e guardam essa informação na tabela *Receiver Buffer* mas, o nó 3 não recebe (possivelmente devido às interferências com outros nós vizinhos). O nó 5, ao receber o ACK dos vizinhos que receberam correctamente a trama de dados, guarda a trama na tabela *Send Buffer*, e elege o nó 3 como o próximo vizinho para transmitir. Após executar a fase de *collision avoidance*, o nó 5 envia a trama RTS com uma nova gama do número de sequência 0 – 1, ou seja, a primeira trama já foi transmitida e com a sequência de valor 0, sendo posteriormente enviada a trama corrente que correspondente à sequência 1. Assim que o nó 3 recebe a trama RTS, ao examinar a sua tabela de *Receiver Buffer* nota que ainda não recebeu a trama da sequência 0. O nó 3 envia CTS com essa indicação, ou seja, solicitando a trama da sequência 0. O nó 5, ao receber a trama CTS, copia da tabela *Send Buffer* a trama de dados correspondente à sequência 0, e envia-a para o nó 3. Depois de receber a trama de dados, o nó 3 actualiza a sua tabela *Receiver Buffer* e responde com um ACK. Ao receber o ACK, o nó 5 envia novamente outra trama RTS com a gama de número de sequência 0 - 1. O nó 3, ao receber a trama RTS, verifica a tabela *Receiver Buffer* e nota que já tem a trama de sequência 0, mas ainda não tem a trama da sequência 1. Então, envia uma trama CTS solicitando trama com a sequência 1. Após receber a trama CTS, o nó 5 envia a trama de dados correspondente à sequência 1. Por sua vez, o nó 3, ao receber a trama de dados, responde com um outro ACK.

O protocolo *BMW* é fiável porque o emissor retransmitirá uma trama de dados quantas vezes for necessário até receber um ACK dos seus receptores, ou seja, com este procedimento consegue garantir que todos os nós de destino receberão as tramas que lhes são destinados. Mas por outro lado, este tipo de comportamento introduz alguma ineficiência de desempenho devido às razões já conhecidas:

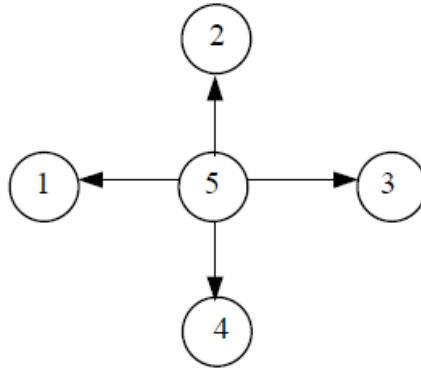


Figura 2.3: O nó 5 efectua *broadcast* da trama aos nós vizinhos.

- *contention phase*: Necessita de pelo menos uma fase de contenção de cada vez que precisa de enviar uma trama. Como o acesso ao meio terá de ser justo (oportunidades de acesso semelhantes para todos os nós), o nó emissor também terá de se conter perante os outros nós para o acesso. Dessa forma, os outros nós podem ter acesso ao meio, interrompendo o procedimento de *multicast*.
- *Timeout*: em muitas aplicações em que se usa *multicast*, se o pedido *multicast* não for realizado dentro de um determinado período de tempo as camadas acima do MAC podem considerar o pedido como inválido, fazendo com que seja sempre atingido o *Timeout* e, conseqüentemente, irão gerar mais tramas na rede diminuindo o débito do sistema.

2.4 BMMM e LAMM

No trabalho apresentado em [12], foram propostos dois protocolos: ***Batch Mode Multicast MAC protocol*** (*BMMM*), que consiste basicamente na redução das fases de contenção dos nós perante os seus n vizinhos para uma única fase de contenção e o protocolo ***Location Aware Multicast MAC protocol*** (*LAMM*), que usa a informação da localização dos nós para melhorar o protocolo *BMMM*.

No protocolo **BMMM**, o nó que possui a trama de dados, e que pretende transmi-

tir terá de ter a certeza que não haverá colisões após o seu envio. Dessa forma, surge a necessidade de fazer uma coordenação das tramas de controlo RTS, CTS e ACK. O nó emissor quando envia uma trama RTS, envia já a instrução de como é que os receptores irão enviar as tramas CTS. Isto, porque em *Multicast* o período SIFS é igual para todos os nós, e assim que recebem a trama RTS e após a duração SIFS, poderá suceder que vários nós enviam trama CTS em simultâneo, originando uma colisão no emissor. Para coordenar a recepção das tramas ACK, acrescenta-se mais uma trama de controlo que é RAK (*Request for ACK*). Com o envio periódico das tramas RTS e RAK antes e depois das tramas de dados, respectivamente, previne-se que qualquer vizinho do nó que envia trama de dados passe a sua fase de contenção.

Analisando o resultado das simulações do trabalho descrito em [12], verifica-se que o ritmo de entrega dos pacotes com sucesso varia inversamente com o número de nós no meio partilhado. Isto, deve-se ao facto da probabilidade de ocorrerem transmissões em simultâneo aumentar, e consequentemente, os pacotes podem atingir facilmente o tempo máximo (*timeout*) em que podem circular na rede até chegarem ao nó de destino e serem descartados. Note-se que o *timeout* tem um valor importante na definição da fiabilidade, pois quanto maior o valor de *timeout*, maior será o ritmo de entrega com sucesso.

O trabalho [12], propõe claramente a redução do número das fases de contenção dos nós, diminuindo dessa forma o tempo total que uma dada trama de dados necessita para alcançar ao seu destino. Deste modo, reduz-se também a probabilidade duma trama de dados atingir o *timeout*. Contudo, estes protocolos não apresentam bom desempenho em termos de justiça de acesso ao meio entre os nós, e não propõem uma solução para os problemas inerentes à rede multi-hop (secção).

2.5 Esquema de *broadcast* duplicado

No protocolo IEEE 802.11, cada trama *broadcast* é transmitida uma única vez, ou seja, as tramas não são retransmitidas. A inundação na rede (denominado por *flooding fraction*), pode não ter uma cobertura que abrange todos os nós na rede. Portanto, uma das formas

simples de aumentar a área de inundação consiste em aumentar o número das tramas *broadcast* enviadas por cada nó, e se cada nó transmitir tramas *broadcast* duas vezes, a área da inundação na rede será ainda mais reforçada do que realizando uma única transmissão. Do ponto de vista da rede, transmitir muitas tramas *broadcast* iguais pela rede, não é a melhor opção pois gera tráfego desnecessário na rede, que tem como consequência a diminuição do débito do sistema, para além de poder originar o fenómeno de *broadcast storm* [13].

Duplicação de *broadcast*

No trabalho apresentado em [14], foi proposto um esquema para evitar que um nó transmita tramas *broadcast* que já foram recebidas por todos nós vizinhos, ou seja, que são redundantes. Só é necessário retransmitir a trama *broadcast* se houver algum nó vizinho que não recebeu anteriormente. Para atingir este objectivo, é necessário obter duas informações importantes dos nós que enviam tramas *broadcast*: o número dos nós vizinhos activos e o número dos vizinhos que receberam correctamente a trama *broadcast*. Essa informação pode ser determinada mantendo uma tabela de conectividade local - (LCT) em cada nó. Cada vez que um nó recebe uma trama, este actualiza a LCT de acordo com o endereço de origem daquela trama, (LCT contém o número dos nós vizinhos de um determinado nó).

Como um dos grandes problemas das transmissões de tramas *broadcast* é causado fundamentalmente pela falta de um mecanismo do reconhecimento (ACK) na transmissão e, para assegurar o estado das tramas *broadcast* enviadas, os autores modificaram ligeiramente o protocolo MAC IEEE 802.11 para que o mesmo possa ter o *Acknowledgment*. Para evitar o *overhead*, o esquema é desenhado de forma a que todos os receptores respondam de imediato dentro do respectivo intervalo DIFS. Aplicando o mesmo procedimento de *collision avoidance* em CSMA/CA, o período DIFS de $50\mu s$ (que neste esquema, é denominado como *Backoff Acknowledgement window* (BACK W)) é dividido em *minislots* e, cada nó receptor irá seleccionar um deles para transmitir o ACK, tal como apresentado na Figura 2.4. Como o tempo de transmissão do pacote ACK é maior do que o intervalo DIFS, os *Acknowledgment* de tramas *broadcast* (BACK) devem ser suficientemente

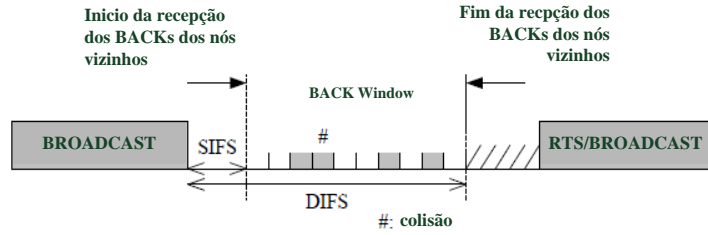


Figura 2.4: Ilustração de esquema BACK.

pequenos para estarem contidos nos *minislots* do *BACK_Window*, evitando assim serem confundidos com o ACK das tramas de dados. Para ajudar os nós a reconhecer as mensagens BACK, os autores usaram um padrão, x , de m -bit para identificar as mensagens BACK. Desta forma, foram usados dois padrões $p(x)$ neste esquema:

$$p(x) = \begin{cases} 0, & \text{novo ACK numa trama } broadcast \text{ recebido} \\ 1, & \text{ACK duplicado em relação a uma trama } broadcast \text{ recebido} \end{cases} \quad (2.3)$$

Estes dois padrões foram desenhados para informar o nó emissor da recepção numa trama *broadcast* nova ou duplicada. Assim que um receptor recebe uma trama *broadcast*, escolhe aleatoriamente um *BACK minislot* para preencher o padrão correspondente. Para um padrão de x -bits numa *WLAN* de m -Mb/s, $\frac{DIFS-SIFS}{x/m} = \frac{50-10}{x/m}$ *minislots* serão definidos em termos de *BACK_W*. Por exemplo, se se usar um padrão de 4-bits numa rede *WLAN* de 11 Mb/s, são utilizados 110 *minislots*, ou seja, *BACK_W* = 110.

E assim, de acordo com o esquema *broadcast* de *Acknowledgment*, qualquer nó intermediário terá todas as informações para saber se deve fazer o *rebroadcast* da trama ou não. Basicamente a primeira lógica aqui aplicada é verificar se o número de BACK recebido é menor do que o número total de vizinhos menos 1 (*LCT* - 1). Se assim for, o nó faz o *rebroadcast* da trama porque sabe que existem vizinhos que não confirmaram a recepção do *broadcast* com o BACK. Definiu-se ainda um limite máximo para tentativas de *broadcast* *MBRT* (*maximal broadcast retry threshold*) para evitar o desperdício de largura de banda.

As tramas do *broadcast* terão uma prioridade maior do que tramas *unicast*. Quando

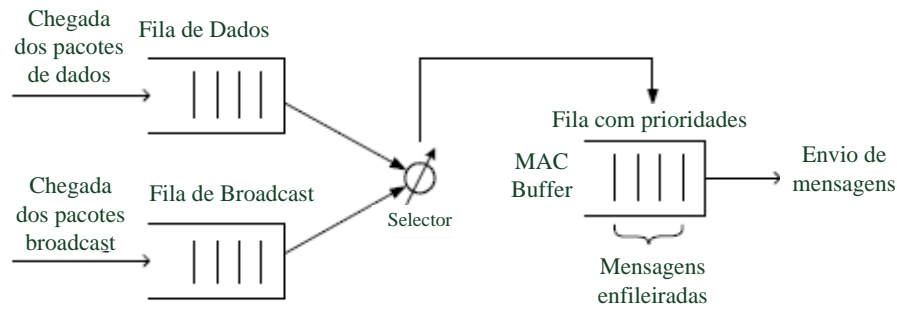


Figura 2.5: Modelo de enfileiramento dos pacotes de dados e broadcast na camada de rede e fila de prioridade na camada MAC.

os pacotes chegam da camada de rede, estas são separadas em duas filas diferentes (ver Figura 2.5). Além disso, para evitar que as tramas *broadcast* circulem de uma forma desnecessária na rede, cada trama *broadcast* deverá conter os seguintes campos:

- Source Address (SA)
- Destination Address (DA)
- broadcast ID (BID)
- Hop count
- Retry flag
- Data Payload

O par $\langle SA, BID \rangle$ serve para identificar as tramas *broadcast* e assim, para a mesma trama dá mais prioridade àquela com maior *hop count*, que representa o estado de um contador e que é incrementado de cada vez que um pacote passa por um nó. O campo *Retry flag*, assim como o nome indica, é uma *flag* que serve para identificar se a trama é nova, ou uma mera retransmissão. Adicionalmente, os nós armazenam dois contadores, *BACK count* e *Retry count*. O *Retry count* é usado para guardar o número de tentativas de *broadcast* de uma determinada trama (BID). Se atingir o MBRT, a trama é descartada imediatamente. O *BACK count* serve para controlar quantas tramas BACK são esperadas dos nós vizinhos, antes que a trama *broadcast* seja descartada, e é normalmente iniciado com o valor LCT (Número total dos vizinhos activos), atingindo o valor 0 após todos os nós vizinhos terem

transmitido a trama BACK.

Com este trabalho, verifica-se uma maior cobertura de transmissão mesmo em situações com grande densidade dos nós no meio, porque quando uma trama *broadcast* falha num determinado nó, existe elevada probabilidade da mesma trama ser recuperada noutro nó sem a necessidade de fazer *rebroadcast* do mesmo. Nos resultados apresentados na Figura 8 do trabalho [14], observa-se que o protocolo IEEE 802.11 com duas filas de espera obtém melhor desempenho do que no caso em que só se utiliza uma fila única (tal como proposto na norma IEEE 802.11).

2.6 LBP, DBP e PBP

O trabalho [15], tem como objectivo a entrega fiável dos pacotes *multicast* aos nós receptores (*multicast*). Foram propostos três protocolos para atingir este objectivo, que são: *Leader Based Protocol* - LBP que consiste em eleger um nó líder do grupo, *Delayed Feedback Based Protocol* - DBP baseando em acesso ao meio através de temporizadores aleatório e ainda o *Probabilistic Feedback Based Protocol* - PBP onde o acesso ao meio é com base nas medidas probabilísticas.

LBP consiste em eleger um dos nós receptores como líder do grupo, sendo este o nó responsável pelo envio de tramas CTS e ACK em resposta às tramas RTS e tramas de dados que são enviadas pelo nó central (com funcionalidades idêntica a uma estação base) respectivamente.

No DBP não existe um nó líder para coordenar o envio de tramas CTS. Os nós receptores respondem às tramas RTS, de acordo com um temporizador inicial que é escolhido aleatoriamente dentro duma gama $\{1, 2, \dots, L\}$. Este L é fornecido pelo emissor explicitamente nas tramas RTS. Só ocorre colisão no emissor se dois nós receptores escolherem o mesmo temporizador para o envio de tramas CTS. O nó em que o temporizador expire envia a trama CTS, e ao ouvir o CTS, os outros nós membros que ainda não têm temporizador expirado colocam os seus temporizadores em pausa. Como os receptores enviam tramas CTS com um determinado atraso, a estação base deve esperar um período de tempo para ouvir a trama CTS. Este período de tempo é o *timeout* da estação base em

T slots. Se a estação base não ouvir a trama CTS dentro deste tempo T , esta assume que houve uma colisão no meio e entra no estado de contenção, e volta a enviar a trama RTS novamente.

No PBP não existe um nó líder e, os nós receptores não escolhem um número de *slot* aleatório para enviar tramas CTS. Após a recepção da trama RTS, os nós receptores enviam de seguida uma trama CTS com uma probabilidade p . Cada nó no meio tem uma probabilidade p distinta um do outro. Esta probabilidade é escolhida consoante os nós membros no meio. A estação base espera por um *slot* após enviar a trama RTS e se não ouvir a trama CTS naquele segundo *slot*, assume que houve uma colisão no meio e recomeça o processo enviando a trama RTS - *multicast*.

Para a eleição do nó líder, existe uma tabela armazenada na estação base, onde tem o nome de um grupo *multicast* e o seu respectivo líder caso haja um. Quando um nó (n) envia uma mensagem do tipo *join-group* para se juntar ao grupo, a estação base verifica a tabela para descobrir se naquele grupo já existe um nó líder ou não. Se existir, e o nó n não é o líder, a estação base responde que o nó n não será líder daquele grupo. Pelo contrário, se não tiver nenhum nó líder, então o nó n será o novo líder daquele grupo. Quando o nó n deseja deixar de ser o líder envia uma mensagem do tipo *leave-group* para estação base e então será eliminado da tabela. Mas, se o nó n enviar a mensagem para deixar de ser o líder do grupo, e a estação base não ouvir a mensagem devido à alguns factores tais como a mobilidade, as interferências entre os nós, entre outros, a estação base pode continuar a assumir que aquele nó n ainda é líder do grupo. A partir daí não receberá mais tramas CTS daquele nó n até o eliminar da tabela. Como consequência disso o tráfego gerado de controlo diminuirá o débito do sistema.

De acordo com os resultados analíticos do trabalho [15], o LBP tem um melhor desempenho do que os protocolos DBP e PBP. Contudo, o LBP apresenta algumas dificuldades quando se trata de nós com alta mobilidade, porque só o nó líder é que tem a permissão de enviar tramas CTS e quando o actual deixar de o ser, terá de ser escolhido um novo líder para aquele grupo específico, o que não é muito consistente em redes móveis. No trabalho apresentado em [15], encontra-se informação mais detalhada acerca do funcionamento destes protocolos.

2.7 Dual Busy Tones

Neste trabalho [1], foi proposto uma extensão ao protocolo MAC IEEE 802.11 para melhorar o desempenho da transmissão de pacotes *multicast* em cenários de comunicações sem fios. Como uma antena não consegue efectuar o *sensing* do canal enquanto está a transmitir, os autores admitiram que cada nó tem dois emissores rádio, um é usado para enviar e receber os tons (*busy tones*), enquanto o outro é usado para transmitir e receber tramas de dados e tramas *NAK tones*.

Definições de alguns termos para este protocolo:

- **Not-ready Response Time:** (NRT): é o tempo em que um nó pode enviar um tom para indicar que não está disponível para receber qualquer trama de dados, ou então, que a última trama recebida continha erro.
- **Tone:** um tom é um sinal sinusoidal de banda estreita. É transmitida numa frequência diferente do canal que é usado para transmitir tramas de dados, ou seja, um sinal fora da banda.
- **Busy Tone:** é uma sinalização transmitida pelos nós indicando que naquele instante de tempo estão ocupados com outra comunicação. Qualquer nó, só transmite para o meio se não sentir nenhum sinal *busy tone*.
- **NCTS Tone:** este tom é transmitido para responder a uma trama RTS, indicando que naquele instante, o nó receptor em questão não está disponível para receber tramas de dados, ou que a trama RTS recebida contém erro.
- **NAK Tone:** este tom é transmitido pelos nós que recebem tramas de dados com erro.

Descrição do algoritmo

O algoritmo é descrito num diagrama de estados (ver Figura 2.6). Quando um nó pretende transmitir qualquer dado (S1), verifica o canal. Se o canal estiver livre por um período DIFS, o nó resume o seu período de contenção. Se o canal continuar livre quando o período de contenção expirar, o nó pode começar a transmitir. Para uma comunicação do tipo

unicast, o nó envia uma trama RTS e espera uma outra trama CTS (estado $S2 \rightarrow S3 \rightarrow S5$). Se o nó emissor receber a trama CTS, envia a trama de dados e espera por um ACK (estado $S5 \rightarrow S7$). Se o nó emissor receber o ACK assume-se que a transmissão dos dados teve sucesso (estado $S1$). Se o nó tiver mais dados para a transmissão repete toda a operação descrita.

Quando os dados a transmitir são do tipo *multicast*, o acesso ao meio é feito da mesma forma que em *unicast*. Mas, quando envia tramas RTS (estado $S2 \rightarrow S3 \rightarrow S4$), o nó emissor não espera a recepção das tramas CTS da parte dos nós receptores. O nó emissor observa o canal de sinalização para verificar se nenhum nó naquele instante está a enviar um tom do tipo NCTS. Se não for detectada nenhuma sinalização NCTS, o nó emissor começa a transmitir pacotes do tipo *unicast* (estado $S4 \rightarrow S8$). No final da transmissão o nó emissor verifica o canal de sinalização novamente para detectar sinais do tipo NAK, e se for oculta o emissor assume que a transmissão teve sucesso (estado $S8 \rightarrow S1$).

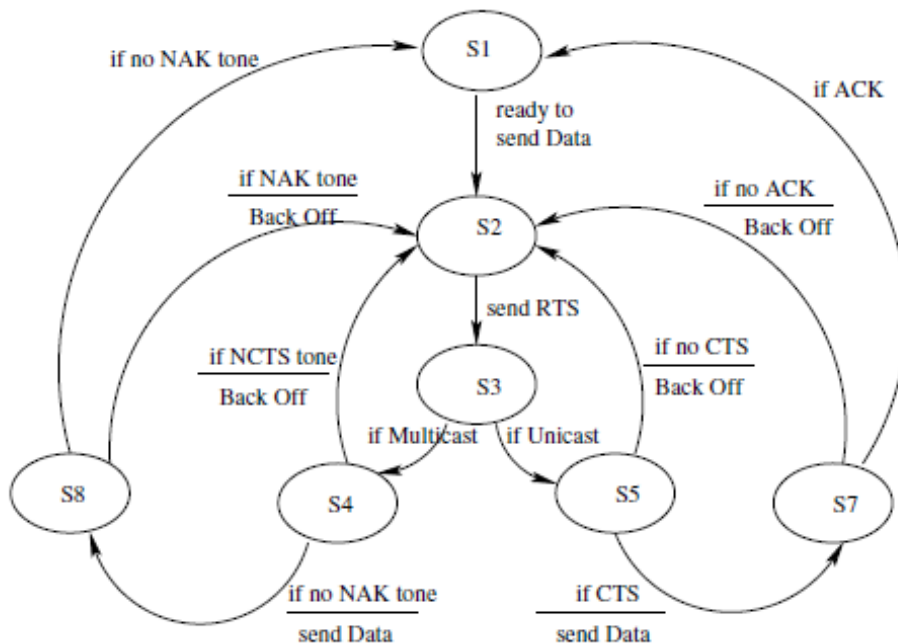


Figura 2.6: Diagrama de estados do emissor (tirada do [1]).

No lado do receptor, o procedimento é semelhante, só que no sentido inverso (ver Figura 2.7).

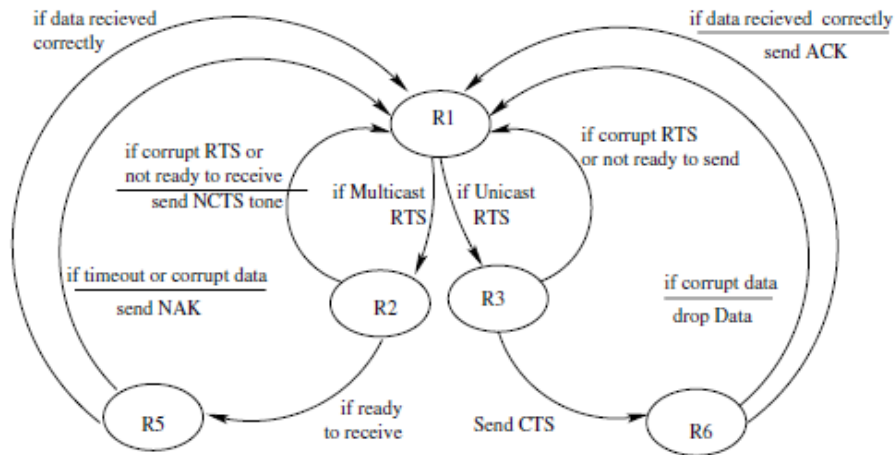


Figura 2.7: Diagrama de estados do receptor (tirada do [1]).

Incorporação de *Dual Busy Tones*

Um nó que não esteja no raio de alcance de uma conversa o iniciada entre um par de n s, pode n o ouvir as suas trocas de tramas RTS e CTS. Este n o ao entrar para a zona de comunica o dos outros pares dos n s, pode fazer o *sensing* do canal durante o longo sil ncio provocado entre uma trama RTS e a trama de dados (no caso de transmiss o *multicast*). Este sil ncio pode erradamente levar o n o a assumir que o meio est  livre e pode come ar a transmitir, causando uma colis o. O n o receptor assim que detectar a transmiss o, envia logo uma sinaliza o do tipo NAK. Mas, com esta t cnica proposta pelos autores em [1], pode-se prevenir este constrangimento usando *Dual Busy Tones*.

Algumas considera  es finais

No trabalho [1] s o usados tons ao inv s de pacotes para sinalizar NAK. O alto d bito e a fiabilidade atingidos, adv m dos transceivers adicionais utilizados pelos n s. Um transceiver   necess rio para lidar com o *Dual Busy Tones* enquanto o outro para DATA/NCTS/-NAK. Mas com a incorpora o de mais transceivers o custo do sistema aumenta, pois ser  necess rio fazer uma altera o a n vel do hardware dos terminais e com duas antenas o consumo da bateria   maior.

2.8 MyopicMAC

No trabalho [2], e assim como no protocolo IEEE 802.11, a coordenação dos nós é distribuída e baseada no protocolo CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) para aceder ao meio. Mas, ao contrário do IEEE 802.11 em que a janela é duplicada sempre que há uma colisão, [2] considera que os nós usam uma janela de contenção com valores próximos do valor óptimo. Os nós usam sempre uma janela de contenção não dependente do estado da transmissão anterior (sucesso ou colisão) como é o caso de BEB em IEEE 802.11. O foco aqui é ter a fiabilidade e um débito elevado.

A probabilidade de um nó transmitir num determinado *slot* escolhido aleatoriamente, é dada por τ , em que depende apenas do tamanho da janela de contenção (Cw).

$$\tau = \frac{2}{Cw + 1} \quad (2.4)$$

O débito S do sistema normalizado é dado pela fracção do tempo em que o canal foi usado com sucesso para transmissão de dados. Para calcular o S é necessário analisar o que pode acontecer a um determinado *slot* aleatório, ou seja, a probabilidade de encontrar *slots* onde o canal possa estar livre, com a probabilidade

$$P_i = (1 - \tau)^n, \quad (2.5)$$

ou canal ocupado devido a uma transmissão com sucesso

$$P_s = n\tau(1 - \tau)^{n-1}, \quad (2.6)$$

ou ainda, o canal ocupado devido à uma colisão no meio

$$P_c = 1 - P_s - P_i. \quad (2.7)$$

Assim, o débito do sistema é dado como o rácio entre a quantidade do tempo que o canal foi usado com sucesso e o tempo total em que o canal foi usado.

$$S = \frac{P_s \mathbb{E}[p]}{P_i \mathbb{E}[T_i] + P_s \mathbb{E}[T_s] + P_c \mathbb{E}[T_c]}, \quad (2.8)$$

onde $\mathbb{E}[T_s]$, $\mathbb{E}[T_c]$ e $\mathbb{E}[T_i]$ são vistos como a duração de uma transmissão com sucesso, colisão e *slot* livre, respectivamente. $\mathbb{E}[p]$ é a duração média da trama de dados.

O sistema consegue atingir um débito próximo do débito óptimo S^* se cada nó no meio tiver a probabilidade de acesso ao canal óptimo τ^* , o que pode ser escrito em função f (definido pelos autores) do número dos nós

$$\tau^* = f(n) \quad (2.9)$$

Como a probabilidade de acesso ao meio óptimo τ^* só depende do número dos nós no meio, surge a necessidade de estimar o número dos nós para obter o τ^* .

Estimativa de número dos nós

Durante o período de contenção um nó pode calcular a probabilidade de encontrar um *slot* livre, dada por \tilde{p}_a , ao observar o estado dos *slots*. Se o sistema estiver em operação num estado perto do ponto de equilíbrio, todos os nós adoptam a probabilidade de acesso ao meio óptimo τ^* . Quando isso acontecer um nó só observa um *slot* livre no meio se os restantes nós $(n-1)$ não transmitirem naquele instante, significa isso que $p_a = (1 - \tau^*)^{(n-1)}$. Desta forma a estimativa do número dos nós no meio é dada pela expressão (2.10).

$$\tilde{n} = \operatorname{argmin} \sqrt{(\tilde{p}_a - (1 - \tau_i)^{n-1})^2}, n \in N \quad (2.10)$$

Implementação do protocolo MAC

Para obter uma baixa variação da medição de \tilde{p}_a , os autores usaram um filtro ARMA (*Auto Regressive Moving Average*). Este filtro tem um comportamento de acordo com a expressão (2.11) onde α representa a memória do filtro, $Slots_i$ é igual a 0 se a i -th *slot* estiver livre, ou igual a 1 se durante i -th o canal estiver ocupado. B é o número dos *slots* observados.

$$\tilde{p}_a(k+1) = \alpha \tilde{p}_a(k) + \frac{1-\alpha}{B} \sum_{i=1}^B SLO T_i, \quad (2.11)$$

Os autores utilizam uma memória com 75% da medição anterior mais o somatório da medição dos restantes 25%, ou seja, $\alpha = 0.75$. Assim, com uma baixa variação do \tilde{p}_a ($\tilde{p}_a(k+1)$ não muito diferente do $\tilde{p}_a(k)$) é possível estimar o número dos nós aplicando a expressão (2.10). De seguida usa-se o (2.9) para calcular a probabilidade de acesso óptimo τ^* . Sabendo que o filtro (2.11) é caracterizado pela sua baixa dinâmica devido ao parâmetro da memória α , o τ^* obtido através de (2.9) deve ser controlado levando em conta a dinâmica do filtro. Para isso, os autores usaram um controlador PI (*Proportional-Integral*) (ver Figura 2.8). O controlador simplesmente adapta a probabilidade de acesso

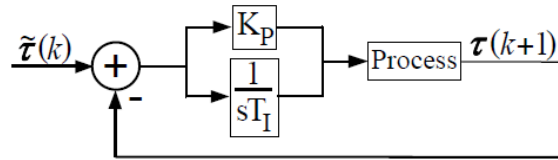


Figura 2.8: Controlador PI $C(\tilde{\tau})$ (tirada do [2]).

individual de cada nó à referência $\tilde{\tau}(k)$ e a dinâmica do filtro ARMA.

Através das Figuras 4(a) e 4(b) de [2], observa-se um melhor desempenho deste protocolo comparativamente ao protocolo IEEE 802.11 à medida que o número dos nós aumentam. No IEEE 802.11 à medida que aumenta o número dos nós e com o aumento das tramas *broadcast* p_b , são geradas mais colisões no meio porque a janela de contenção Cw de valor fixa 32 não está adaptada ao número dos nós. Consequentemente observa-se uma diminuição do débito. Já no [2] note-se que o débito é quase independente do número dos nós. Isto é devido ao ajustamento óptimo da janela de contenção.

A fiabilidade (probabilidade de transmissão com sucesso) referente às Figuras 5(a) e 5(b) do [2], diminui com aumento do número dos nós que provoca o aumento da probabilidade de colisão. Isto porque quando a transmissão é somente de tramas *broadcast* $p_b=1$, não há retransmissão da trama. Se a transmissão falhar, a trama simplesmente é descartada.

2.9 Resumo

Os trabalhos descrito nas secções de 2.3 a 2.6 apresentam algumas modificações do protocolo IEEE 802.11 (secção 2.2) para melhorar a fiabilidade do mesmo em caso das transmissões do tipo *broadcast*. Como foi descrito, alguns trabalhos apresentam como solução a coordenação do envio das tramas CTS e outros a incorporação das tramas *Acknowledge*, solução adoptada na transmissão das tramas *unicast*. O trabalho apresentado na secção 2.8 apresenta um novo esquema para a camada MAC independentemente do tipo das tramas (*unicast* ou *broadcast*) porém, para cenários a um salto (single-hop). É sobre este último que assenta o tema principal desta dissertação, o estudo da fiabilidade da transmissão para cenários *multi-hop*.

2.10 Problemas associados às redes multi-hop

Nó escondido

O problema do terminal (nó) escondido sucede quando um nó se encontra fora do alcance rádio de um emissor mas dentro do alcance do seu receptor. Por exemplo, na Figura 2.9, o nó C está fora do alcance rádio do nó A, pelo que uma transmissão do nó C não será ouvida pelo nó A. Portanto, enquanto o nó C está a transmitir para o nó B, o nó A interpreta que o canal está livre e por conseguinte pode começar uma transmissão para o nó B. Isto faz com que ambas as tramas (do nó C e do nó A) colidam no nó B devido á interferência, e serão vistos como "colisão". Uma transmissão do nó A para o nó B enfrentará as mesmas consequências, mesmo que o nó C esteja a transmitir para um outro nó, como por exemplo o nó D.

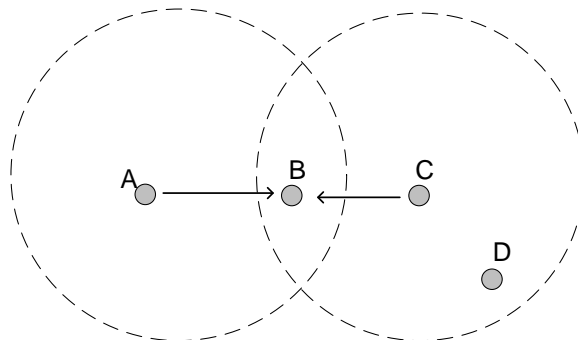


Figura 2.9: Terminal (nó) escondido.

Nó exposto

O problema do nó exposto é o problema no sentido inverso. Este sucede quando uma transmissão ou um nó “exposto” se encontra dentro do alcance rádio do emissor mas fora da do receptor. Por exemplo na Figura 2.10, quando o nó B pretende transmitir para o nó A, encontra o canal ocupado devido ao facto de naquele instante estar a decorrer uma transmissão do nó C para o nó D. Pelo que, o nó B terá de aguardar até que a transmissão do nó C termine para poder transmitir para o nó A, uma vez que não seria necessário porque a transmissão do nó B não interfere no nó D.

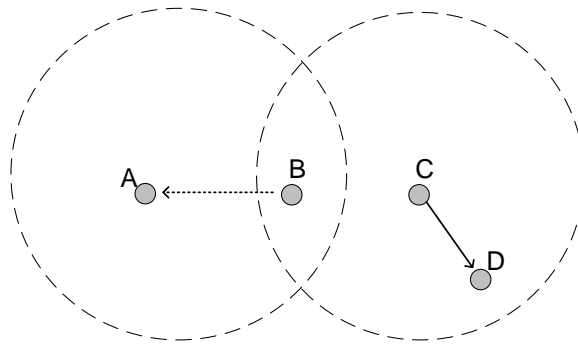


Figura 2.10: Terminal (nó) exposto.

Capítulo 3

MODELO TEÓRICO

3.1 Introdução

Uma das etapas mais importantes no desenvolvimento de um protocolo, com vista a ser utilizado em qualquer tipo de redes, é o seu teste/validação, ou seja, a necessidade de aproximar o mais possível o cenário de teste e de validação ao ambiente real correspondente. Neste capítulo é apresentado um modelo para o protocolo MAC, que serve de base para o desenvolvimento do protocolo apresentado no Capítulo 4.

3.2 Probabilidade de Acesso Individual

Cada estação (nó) calcula um intervalo de tempo aleatório denominado de período de contenção, que é distribuído uniformemente entre 0 e o tamanho da janela de contenção Cw ($Backofftime(BT) = rand(0, Cw) \times TimeSlot$), e o valor do *Time Slot* depende do tipo da tecnologia usada na camada PHY. O contador do período de contenção é decrementado após um período DIFS, enquanto o meio estiver livre (sem nenhuma transmissão no meio), e congela a contagem se for detectada outra transmissão no meio. Quando uma estação durante a fase de decremento do período de contenção atingir o valor 0, transmite para o meio.

O processo de *Backoff* com janela de contenção Cw pode ser representado por uma cadeia de Markov (ver Figura 3.1), sendo χ_0 o estado em que um nó transmite para o

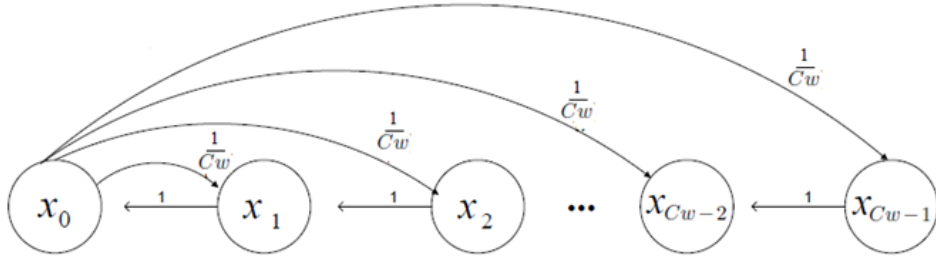


Figura 3.1: Ilustração de uma cadeia de Markov.

meio. A probabilidade do nó estar no estado de contenção χ_{Cw-1} é dado por

$$P_r(\chi_{Cw-1}) = \frac{1}{Cw} P_r(\chi_0), \quad (3.1)$$

e, da mesma forma, a probabilidade de estar em χ_{Cw-2} é dado por

$$\begin{aligned} P_r(\chi_{Cw-2}) &= P_r(\chi_{Cw-1}) + \frac{1}{Cw} P_r(\chi_0) \\ &= \frac{1}{Cw} P_r(\chi_0) + \frac{1}{Cw} P_r(\chi_0) \\ &= \frac{2}{Cw} P_r(\chi_0). \end{aligned} \quad (3.2)$$

Generalizando,

$$P_r(\chi_{Cw-j}) = \frac{j}{Cw} P_r(\chi_0) \quad (3.3)$$

exprime a probabilidade de um nó estar num estado $j = k$, sendo $1 \leq k \leq Cw$. Como a soma da probabilidade do nó estar em todos os estados (0 até $Cw - 1$) é 1, então

$$\sum_{k=0}^{Cw-1} P_r(\chi_k) = 1, \quad (3.4)$$

e, pela progressão aritmética $\sum_{k=1}^n a_k = \frac{n}{2}(a_1 + a_n)$, então

$$P_r(\chi_0) + \sum_{k=1}^{Cw} P_r(\chi_k) = 1. \quad (3.5)$$

Usando a progressão aritmética e a expressão (3.3)

$$\begin{aligned}
 P_r(\chi_0) + \frac{Cw-1}{2} [P_r(\chi_1) + P_r(\chi_{Cw-1})] &= 1 \\
 P_r(\chi_0) + \frac{Cw-1}{2} \left[\frac{Cw-1}{Cw} P_r(\chi_0) + \frac{1}{Cw} P_r(\chi_0) \right] &= 1 \\
 P_r(\chi_0) \left[1 + \frac{Cw-1}{2} \right] &= 1 \\
 P_r(\chi_0) &= \frac{2}{Cw+1}.
 \end{aligned} \tag{3.6}$$

E assim, pode-se expressar a probabilidade $\tau = P_r(\chi_0)$ de um nó transmitir num *slot* como

$$\tau = \frac{2}{Cw+1}. \tag{3.7}$$

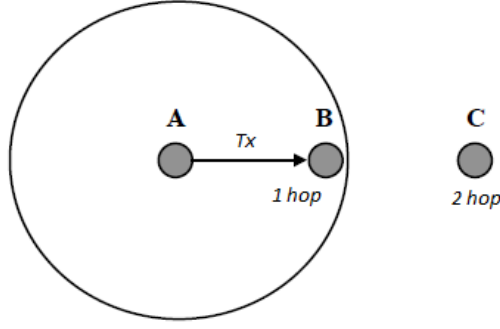
A expressão (3.7) representa a probabilidade individual de cada nó aceder a um *slot* considerando que os nós possuem sempre tramas para transmitir [16].

3.3 Cenário Multi-hop

Nas redes sem fios, uma transmissão pode não ter uma cobertura de rádio que abrange todos os nós na rede e podem ser necessários múltiplos saltos (*hops*) para efectuar a troca da informação entre os nós, daí o termo rede **Multi-hop**. A rede Multi-hop consiste em usar dois ou mais saltos de um nó origem até ao nó de destino através dos nós intermédios (cuja função é de transmitir informação de um nó para outro) para entregar informação. Para saber a probabilidade de uma transmissão ter ou não sucesso é necessário fazer a análise dessa transmissão tendo em conta a possível existência de nós escondidos.

Em cenários de um salto (*single-hop*), ou seja, no raio de alcance de uma transmissão, ilustrado na Figura 3.2, durante a transmissão do nó A para o nó B (um *hop*), se o nó C (escondido do nó A) também transmitir, causará uma colisão.

Definindo $n1$ como o número médio dos nós que constituem a rede a 1 *hop* e $n2$ como o número médio dos nós que constituem a rede a 2 *hops* pode-se analisar o comportamento do MAC a *single* e a *multi-hop*.

Figura 3.2: Cenário *single hop*.

Single Hop

A probabilidade de encontrar um *slot* livre no meio é dada pela probabilidade de nenhum dos $n-1$ nós transmitir num *slot*, e é definida por:

$$P_i = (1 - \tau)^{n-1}. \quad (3.8)$$

No decorrer de uma transmissão do nó A, esta fica exposta no nó B K_e *slots*, em que pode sofrer uma colisão devido ao facto do nó C (escondido do nó A) iniciar uma transmissão. K_e representa a duração da transmissão em número de *slots*, e é dada por

$$K_e = \frac{\mathbb{E}_{DATA} + DIFS + SIFS + 2\delta}{\sigma} \quad (3.9)$$

em que, \mathbb{E}_{DATA} é o tempo médio da duração da trama de dados, δ o atraso de propagação e σ representa o *time slot*. DIFS e SIFS são tempos com a mesma duração definidos na norma IEEE 802.11 [10].

Dois Hops

Se todos os nós têm a probabilidade de transmissão τ , e uma transmissão tiver comprimento de K_e *slots*, então a probabilidade de um nó a dois *hops* (ex: o nó C) aceder ao meio durante a transmissão do nó A pode ser visto da seguinte forma:

- A probabilidade do nó C aceder no primeiro *slot* de transmissão do nó A é dada por

τ

- A probabilidade do nó C aceder no segundo *slot* de transmissão do nó A é dada por $\tau(1 - \tau)$
- A probabilidade do nó C aceder no terceiro *slot* de transmissão do nó A é dada por $\tau(1 - \tau)^2$.

Generalizando, a probabilidade de um nó a 2 *hops* transmitir durante a transmissão é dada por

$$\tau_2 = \sum_{k=1}^{K_e} \tau(1 - \tau)^{k-1}. \quad (3.10)$$

Considerando uma distribuição uniforme de 5 nós no meio, para que um nó possa transmitir com sucesso, é necessário que nem os nós vizinhos localizados a um *hop* e nem os que estão a dois *hops*, transmitam no mesmo *slot*. Caso contrário haverá colisão. Assim, a probabilidade de transmissão com sucesso é dada por

$$P_s' = \tau \times (1 - \tau)^{n_1-1} \times (1 - \tau_2)^{n_2-2}, \quad (3.11)$$

onde a parcela τ corresponde à probabilidade de transmissão de um nó num *slot*, $(1 - \tau)^{n_1-1}$ a probabilidade dos nós vizinhos localizados a 1 *hop* não transmitirem e $(1 - \tau_2)^{n_2-2}$ representa a probabilidade dos nós a 2 *hops* não transmitirem. O expoente $(n_2 - 2)$ exprime o número de vizinhos do nó B que podem interferir com ele próprio pelo facto de iniciarem a transmissão e não ouvirem a transmissão do nó A. Para que uma transmissão

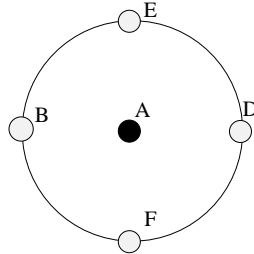


Figura 3.3: Número médio dos nós no raio de transmissão do nó A é 5 (nós A, B, D, E, F).

Multi-hop, com origem no nó A e destino no nó C (ver Figura 3.4) tenha sucesso, é necessário que os nós a um *hop* do nó A (B, D, E, e F) e os que estão a um *hop* do nó B

(C, G e H) não transmitam no mesmo *slot*. Suponha-se um cenário em que o nó A inicia uma transmissão, e fica exposto no nó B K_e *slots*. Se o nó G, embora esteja a dois *hops* do nó A, iniciar uma transmissão em qualquer dos K_e *slots*, haverá colisão no nó B.

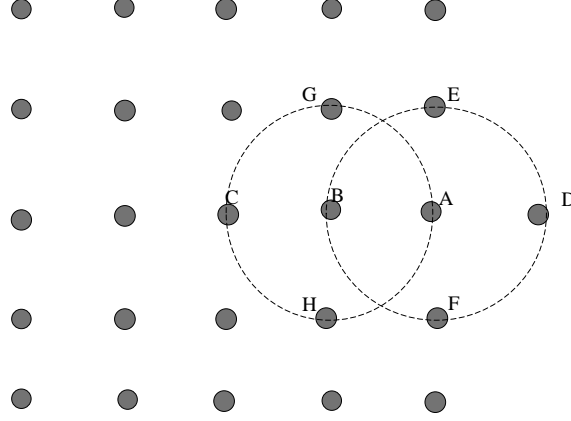


Figura 3.4: Área dos nós contabilizados $n1 - 2$ (C, G e H), para expoente da parcela $(1 - \tau_2)$ na expressão (3.11).

Finalmente, como se consideram um conjunto de $n1$ nós a um *hop*, a probabilidade de sucesso é dada por

$$P_s = n1 \times \tau \times (1 - \tau)^{n1-1} \times (1 - \tau_2)^{n2-2}, \quad (3.12)$$

e a probabilidade de colisão é representada por

$$P_c = 1 - P_i - P_s \quad (3.13)$$

3.4 Débito

Tendo a probabilidade de sucesso e de colisão, pode-se calcular o débito do sistema que é dado por

$$\mathcal{S} = \frac{P_s T_{frame}}{P_i \sigma + P_s T_s + P_c T_c} \quad (3.14)$$

onde T_{frame} é o comprimento médio da *trama*, T_s o tempo médio em que o canal é observado em estado ocupado devido a uma transmissão com sucesso e T_c o tempo médio em que o canal é observado ocupado por cada nó durante uma colisão no meio.

3.5 Validação do Modelo

A validação do modelo apresentado foi realizada através de simulação. Os parâmetros usados para obter os resultados da validação, tanto para análise numérica do modelo descrito como para a simulação, estão descritos na tabela 3.1.

SIFS	10 μs
DIFS	50 μs
EIFS	364 μs
ACK	304 μs
Time slot (σ)	20 μs
Atraso de Propagação (δ)	1 μs
Comprimento das tramas	1500 bytes
ritmo de transmissão de dados	11 Mbps
Mecanismo de transmissão	básico

Tabela 3.1: Parâmetros utilizado para a validação do modelo.

Considerando que todas as tramas de dados tem um tamanho fixo, $\mathbb{E}_{DATA} = \frac{1500.8}{11.10^6} s$, para determinar o débito é necessário calcular os valores de T_{frame} , T_s e T_c , os quais são expressos em *time slot* (σ).

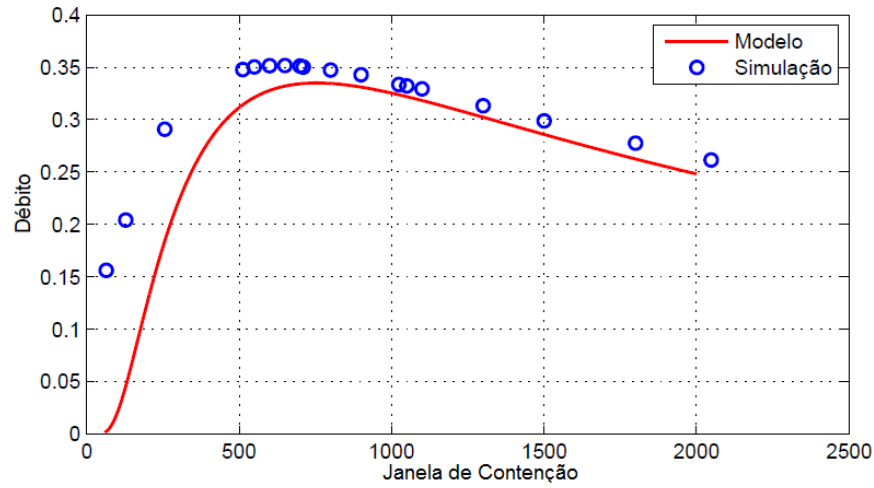
$$\begin{cases} T_{frame} = \mathbb{E}_{DATA} + ACK \\ T_s = \mathbb{E}_{DATA} + DIFS + SIFS + ACK + 2\delta \\ T_c = \mathbb{E}_{DATA} + DIFS + EIFS + \delta \end{cases} \quad (3.15)$$

Para validar o modelo, é necessário comparar os resultados analíticos apurados através do modelo apresentado com os resultados obtidos por meio de simulação, utilizando o simulador ns-2 [9].

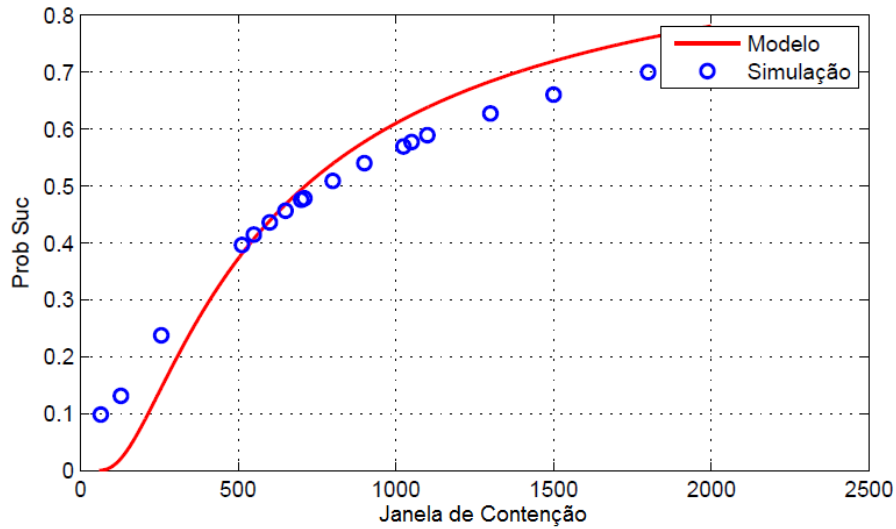
Variando a janela de contenção Cw de 2 a 2000 e com 5 nós no meio ($n1=5$) obtiveram-se os resultados numéricos (modelo), e de simulação (janela fixa) do débito normalizado e da probabilidade de sucesso (ver Figura 3.5).

3.5.1 Notas finais

Os gráficos das Figuras (3.5(a) e 3.5(b)) indicam que o modelo apresentado acompanha a tendência dos resultados obtidos por simulação. Note-se que os resultados para a probabilidade de sucesso são semelhantes. No entanto, para os valores do débito observam-se



(a) Débito



(b) Probabilidade Sucesso

Figura 3.5: Resultados numéricos do modelo

algumas regiões onde o erro do modelo é mais elevado. Os gráficos atingem um pico onde o débito é máximo para uma dada janela (Cw) perto da óptima e, a partir daí começa a baixar novamente como seria esperado. Quando a janela de contenção assume valores elevados, os nós podem ter longos períodos de tempo sem aceder ao canal mesmo que esteja livre, originando um elevado desperdício de utilização do canal, afectando negativamente o débito.

Capítulo 4

PROTOCOLO MAC

Neste capítulo faz-se o uso do modelo teórico apresentado no Capítulo 3, e propõe-se um protocolo MAC que pretende aumentar o débito e a probabilidade de sucesso num cenário multi-hop. No protocolo os nós utilizam uma janela de contenção adaptada de acordo com as estatísticas de ocupação do meio. O protocolo pretende que todos os nós operem com uma janela próxima da óptima, de forma a maximizar o débito da rede.

4.1 Topologia

De forma a não haver interferência com os nós a dois *hops*, foi limitado o *Carrier Sensing* (escuta da portadora) e alcance de transmissão a 111.4 e 100.0 m, respectivamente (ver Figura 4.1). Assim, se dois nós estiverem distanciados de 111.4m, o nó receptor recebe a trama mas não a descodifica. Esta só é recebida e descodificada com sucesso quando a distância a que os dois nós se encontram um do outro seja inferior ou igual a 100m.

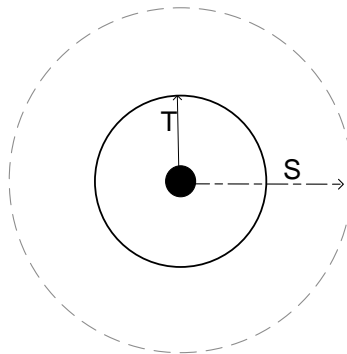


Figura 4.1: T-Alcance de transmissão (100m) e S-Alcance de interferência (111.4m).

4.2 Comparação dos Protocolos Myopic e 802.11

Tendo como motivação a extensão do trabalho descrito na secção 2.8, em que os nós da grelha possuem a fila de espera cheia, ou seja, no estado de saturação, fez-se uma pequena comparação do débito máximo atingido pelo protocolo do trabalho [2] (secção 2.8) que foi implementado para operar em redes a single-hop, com o protocolo da norma IEEE 802.11, num cenário a multi-hop sobre a topologia da secção 4.1. Considerando que no protocolo Myopic não há retransmissão das tramas, este faz com que a janela de contenção (Cw) usada corresponda sempre à janela mínima (Cw_{min}), enquanto que, no protocolo 802.11 a Cw_{min} pode ser aumentada quando as tramas são retransmitidas e o protocolo utiliza o método BEB (secção 2.2). Assim, para cada caso alterou-se o valor de Cw_{min} e registou-se o respectivo débito atingido por cada um dos protocolos.

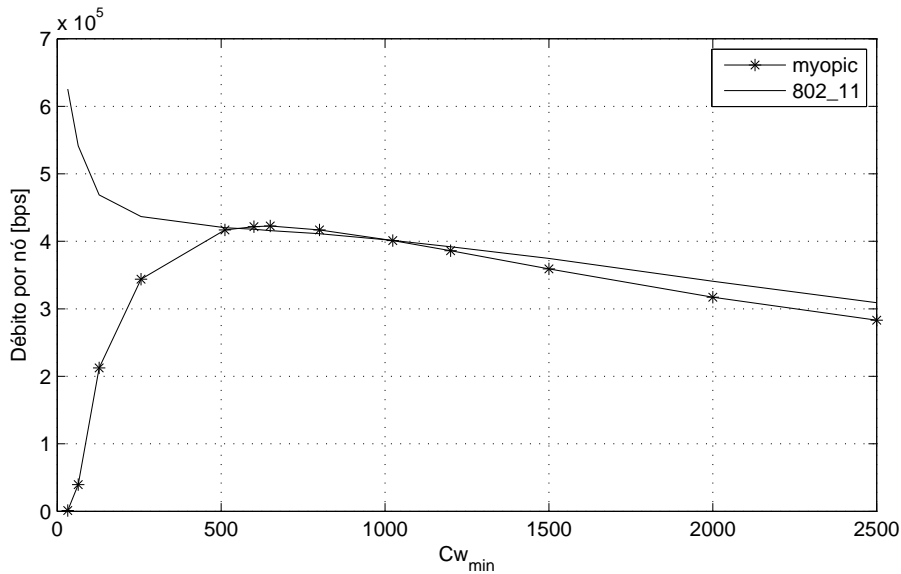


Figura 4.4: Relação de débito dos protocolos MyopicMAC e IEEE 802.11 com a variação do Cw_{min} .

Segundo a Figura 4.4 verifica-se que existe diferença de débito entre os dois protocolos devido à forma de gestão diferenciada da janela de contenção. No protocolo Myopic, todos os nós operam sempre com o mesmo valor de Cw , não havendo retransmissões das tramas. Dessa forma, há maior probabilidade das tramas colidirem e por conseguinte originar menor débito. Enquanto que no protocolo 802.11 como há mecanismo de retransmissões, os nós usam valores de janelas mais diferenciados entre si, permitindo-lhes obter probabili-

dades de acesso mais diferenciadas e diminuindo a probabilidade de colisão, aumentando o débito.

4.3 Análise dos Parâmetros de Optimização

Como se pretende encontrar os parâmetros de funcionamento óptimos para que sejam utilizados no desenvolvimento de um protocolo, primeiro foi verificado numericamente qual a janela que maximizava o débito, para depois com base nela identificar quais os parâmetros a utilizar como referência para o controlo da janela. Considerando a topologia da secção 4.1, assim, ao mudar o valor da janela de contenção dos nós (Cw) analisou-se o débito com a expressão (3.14) e concluiu-se que a janela óptima Cw^* (janela que faz com que o débito atinja o valor máximo) possui sensivelmente um valor de 872 (ver Figura 4.5).

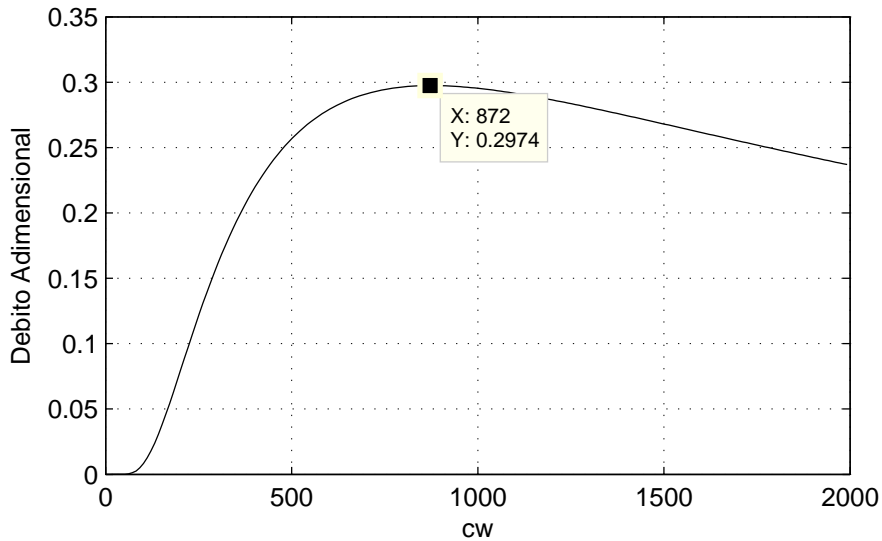


Figura 4.5: Ponto de Cw^* onde se maximiza o débito.

Depois de identificar o valor de Cw^* que maximiza o débito e, tendo em conta a topologia da secção 4.1 em que o número de nós num circulo de cobertura é $n=5$, pode-se calcular analiticamente os valores de P_i , P_s ou P_c . Para o efeito da optimização da janela pode ser usado qualquer um dos parâmetros referidos, quer sejam utilizados de forma individual ou em conjunto. Como P_i e P_s são grandezas que cada nó pode observar no meio, não é necessário trocar qualquer informação entre os nós para as obter, o que as

elege como boas candidatas para o controlo da janela de contenção dos nós.

Através das expressões 3.7 e 3.8 do modelo teórico apresentado no Capítulo 3, obteve-se a expressão para o P_i que um nó observa, como sendo:

$$P_i = \left(1 - \frac{2}{Cw + 1}\right)^{n-1}, \quad (4.1)$$

pelo que, ao variar o valor de Cw também varia o valor de P_i . E, para o valor da janela óptima Cw^* obteve-se o seu respectivo valor de P_i óptimo (P_i^*), ou seja, $P_i^* = \left(1 - \frac{2}{Cw^* + 1}\right)^{n-1}$ com o valor de 0.9909 (ver Figura 4.6).

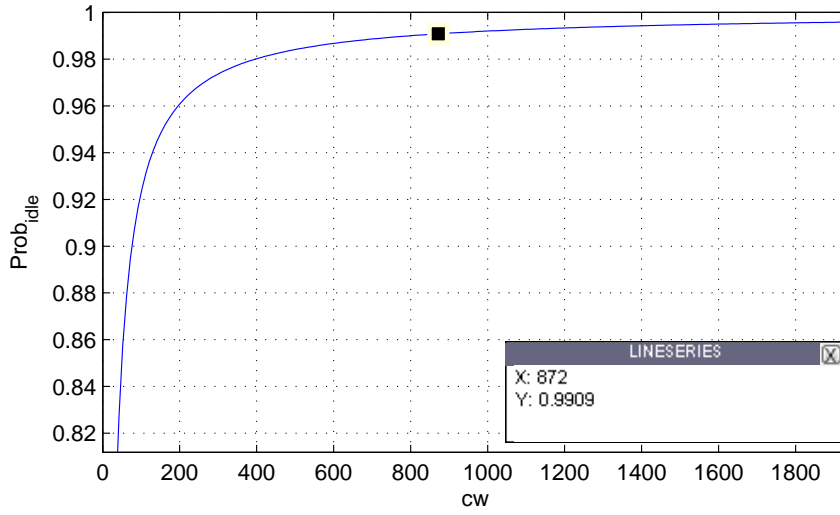


Figura 4.6: P_i óptimo (P_i^*).

Da mesma forma, com a expressão 3.12, obteve-se a probabilidade de sucesso para a janela optima Cw^* (ver Figura 4.7).

Se o protocolo tiver de operar em cenários onde o número médio dos nós na topologia da grelha seja diferente de 5, o valor da janela óptima Cw^* será alterado, e o mesmo se acontecerá para as probabilidades P_i^* e $P_{sucesso}^*$. Este facto motiva a necessidade de analisar qual seria o comportamento dessas grandezas com a variação do número dos nós na rede operando com a janela óptima Cw^* , que varia dependendo do número dos nós (ver Figura 4.8). Com a análise numérica, para cada caso em que se alterou o número dos nós (n) e consoante o procedimento anterior, registou-se os valores de, P_i^* e $P_{sucesso}^*$ na Tabela 4.1.

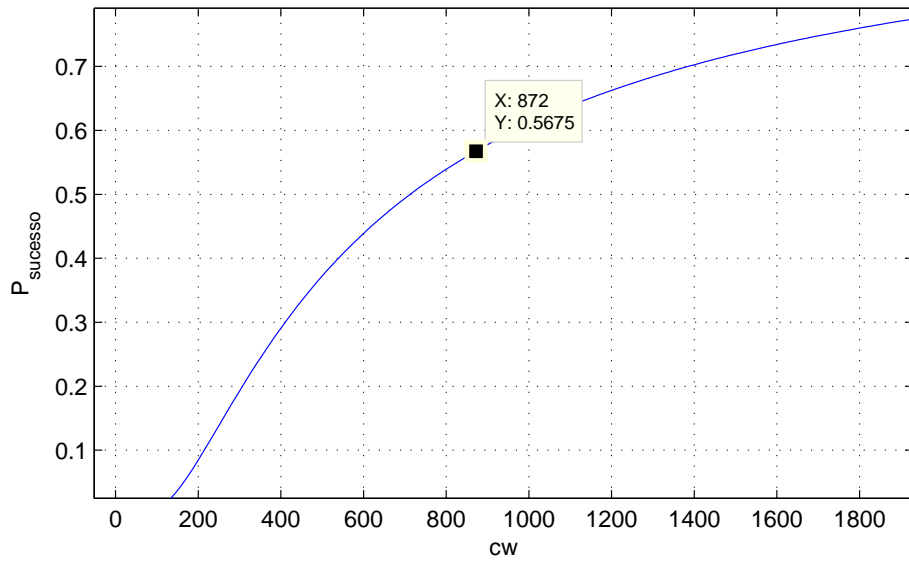


Figura 4.7: $P_{sucesso}$ óptimo ($P_{sucesso}^*$).

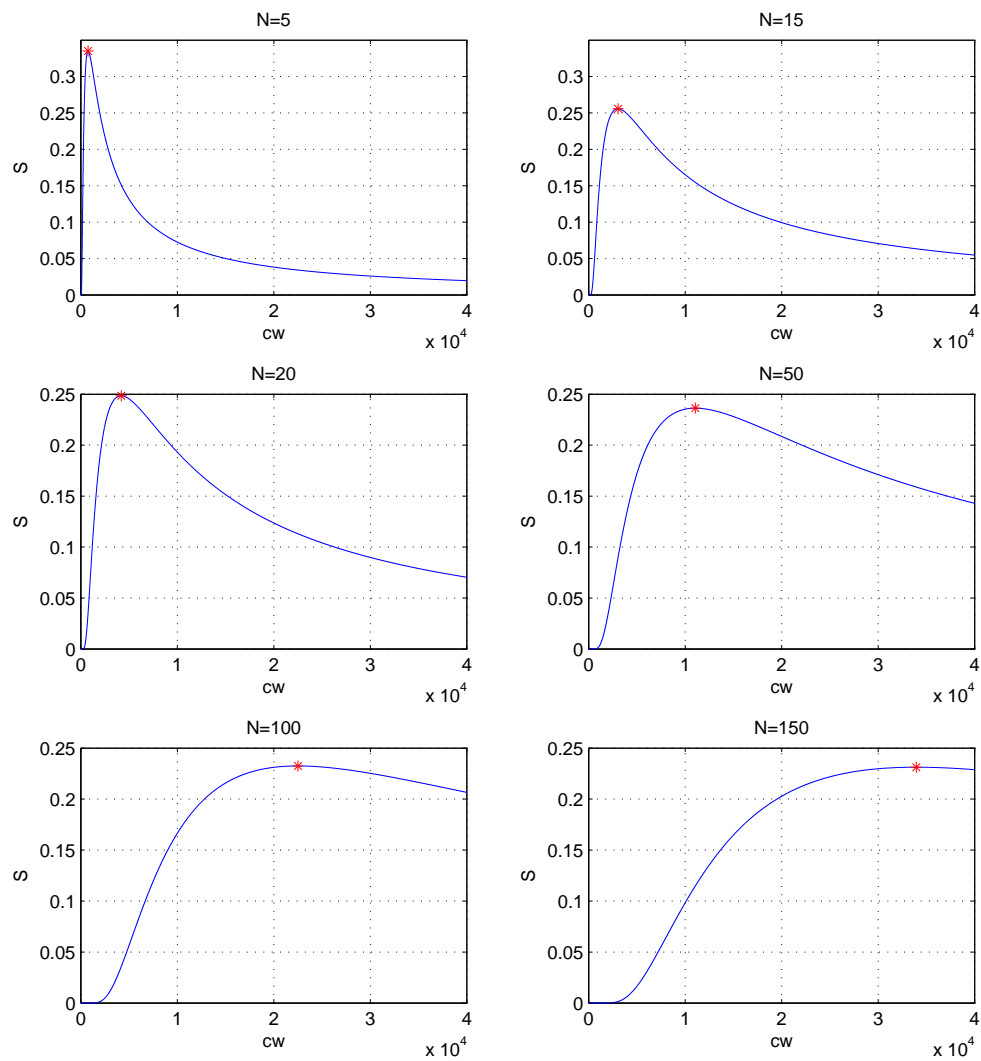


Figura 4.8: Cw^* para diferente número dos nós (n).

n	Cw^*	P_i^*	$P_{sucesso}^*$
10	2302	0.99221	0.56507
15	3602	0.99226	0.55297
20	4952	0.99236	0.55072
25	6302	0.99241	0.54944
30	7652	0.99245	0.54861
35	8952	0.99243	0.54619
40	10302	0.99246	0.54601
45	11652	0.99248	0.54586
50	13002	0.99249	0.54575
55	14302	0.99248	0.5445
60	15652	0.99249	0.54452
65	17002	0.9925	0.54454
70	18352	0.99251	0.54456
75	19652	0.9925	0.54373
80	21002	0.99251	0.5438
85	22352	0.99251	0.54385
90	23702	0.99252	0.54391
95	25002	0.99251	0.54329
100	26352	0.99251	0.54337
105	27702	0.99252	0.54343
110	29052	0.99252	0.5435
115	30352	0.99252	0.54301
120	31702	0.99252	0.54308
125	33052	0.99252	0.54315
130	34402	0.99253	0.54321
135	35702	0.99252	0.54281
140	37052	0.99253	0.54288
145	38402	0.99253	0.54294
150	39752	0.99253	0.54301

Tabela 4.1: Variação do P_i^* com o número dos nós (n).

Prova-se desta forma que o P_i^* e o $P_{sucesso}^*$ se mantêm praticamente constantes ($P_i^* \approx 0.992$ e $P_{sucesso}^* \approx 0.54$) independentemente do número dos nós no meio, conforme é ilustrado na Figura 4.9. Dessa forma, torna-se válido o uso de P_i ou de $P_{sucesso}$ como parâmetros de entrada para controlo da janela de contenção, sendo os valores óptimos de referência $P_i^* = 0.992$ e $P_{sucesso}^* = 0.54$. Para obter o débito óptimo, os nós têm de operar com a janela de contenção adaptável ao meio de forma a terem sempre o P_i e $P_{sucesso}$ mais próximo possível do P_i^* e $P_{sucesso}^*$, respectivamente.

Como a expressão de P_i é amostrada recorrendo somente à informação de ocupação do meio, e é mais simples de obter do que o $P_{sucesso}$, escolheu-se a referência P_i como

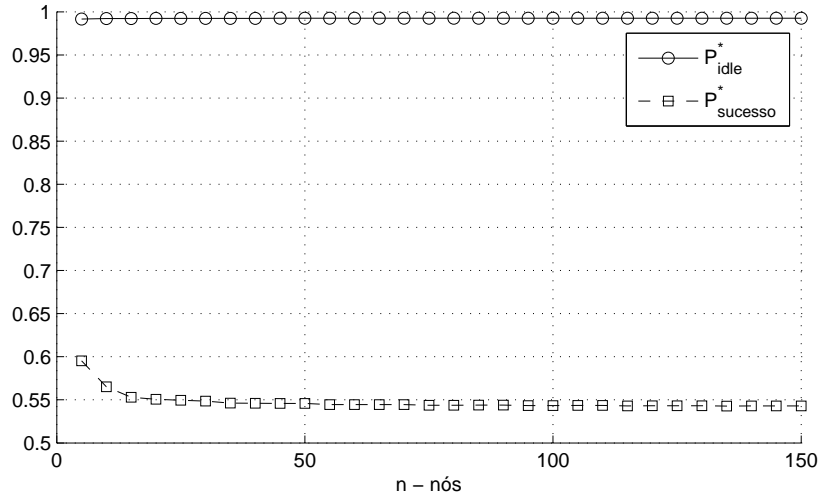


Figura 4.9: Ilustração de probabilidade *idle slots* (P_i^*) e de sucesso ($P_{sucesso}^*$) óptimos com a variação dos nós (n).

parâmetro de observação para regulação da janela de contenção.

4.4 Implementação Prática

Nesta secção é ilustrada a metodologia para o cálculo de probabilidade de sucesso da transmissão e a probabilidade de encontrar *slots* livres.

4.4.1 Probabilidade de sucesso

A probabilidade de insucesso, que é o quociente entre o somatório das tramas enviadas sem sucesso e o somatório das tramas enviadas, é dada por

$$P_{insuc} = \frac{\sum tramas\ falhadas}{\sum tramas\ transmitidas} \quad (4.2)$$

e a probabilidade de sucesso por

$$P_{sucesso} = 1 - P_{insuc}. \quad (4.3)$$

4.4.2 Probabilidade de *idle slots*

Todos os nós, antes de iniciarem qualquer transmissão, entram no estado de contenção. Durante este processo os nós monitorizam o meio de modo a perceberem se está ocupado

ou não (livre). Foi considerado que cada amostragem possui 500 *slots*, assim, um nó consegue-se perceber quantos *slots* estão livres ao fazer a diferença com os ocupados. Como exemplo, supõe-se que durante o estado de contenção um nó percebe-se que 10 *slots* estão ocupados, logo os restantes 490 estão livres. Contudo, a probabilidade de *Idle Slot* é dada por

$$P_i = \frac{Samples - BusySlot}{Samples} \quad (4.4)$$

onde, o *Samples* são os 500 *slots* amostrados inicialmente, e *busy slot* são os *slots* ocupados durante o estado de contenção.

4.5 Optimização da Janela de Contenção

Para obter o débito máximo possível ao longo da operação do protocolo, a janela tem de estar sempre optimizada, ou seja, a probabilidade P_i terá que estar sempre o mais próximo possível do valor da probabilidade óptima P_i^* . Se o meio estiver muito ocupado é porque o P_i tem um valor abaixo do P_i^* . Para efectuar o controlo (aproximar P_i de P_i^*) é necessário aumentar o valor de janela de contenção, que como consequência aumenta também o valor de P_i , e vice-versa.

Para regular a janela de contenção é preciso efectuar a leitura do sistema (adquirir P_i), fazer a comparação de P_i com o P_i^* e só depois actuar (alterar o valor de Cw). Mas como o valor de P_i está sujeito a variações em cada amostragem, foi necessário tomar medidas que minimizem os problemas causados pela variação. Assim, através da simulação, recorrendo ao simulador ns-2 [9] durante 200 segundos e com a janela fixa de valor 850, obteve-se o resultado da média dos valores de P_i medidos durante toda a história da simulação (ver Figura 4.11). Notou-se que existe um ponto a partir do qual a média dos valores de P_i se aproxima de um valor em regime estacionário. Com a identificação deste ponto foi construído um *buffer* circular com a média dos valores de P_i , e o tamanho deste *buffer* corresponde à expressão $\Delta t = t2 - t1$ em *time slots* da Figura 4.10. Tendo em conta as vantagens e as desvantagens associadas ao número de *slots* escolhidos para amostragem, fixou-se o comprimento do universo de amostragem em 100 amostras de 500 *slots*, ou seja, um total de 50000 *slots*. Apenas quando o buffer estiver preenchido é que o sistema

começa a regular a janela de forma a atingir o valor de referência P_i^* , e a partir daí o buffer funciona de forma deslizante.

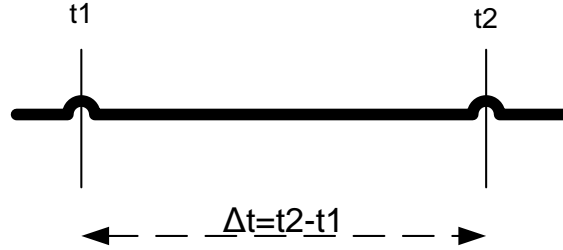


Figura 4.10: Tamanho do *buffer* de $\Delta t = t2 - t1$ *time slots*.

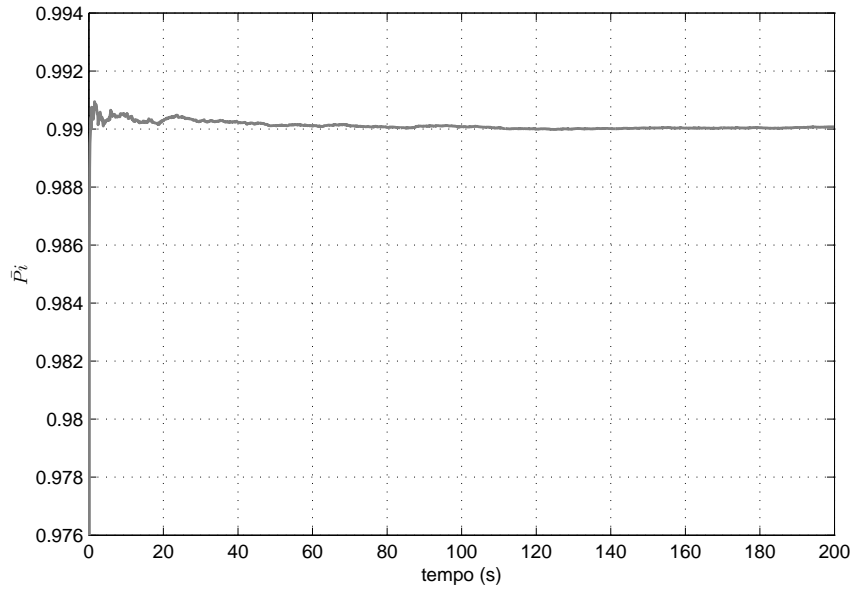
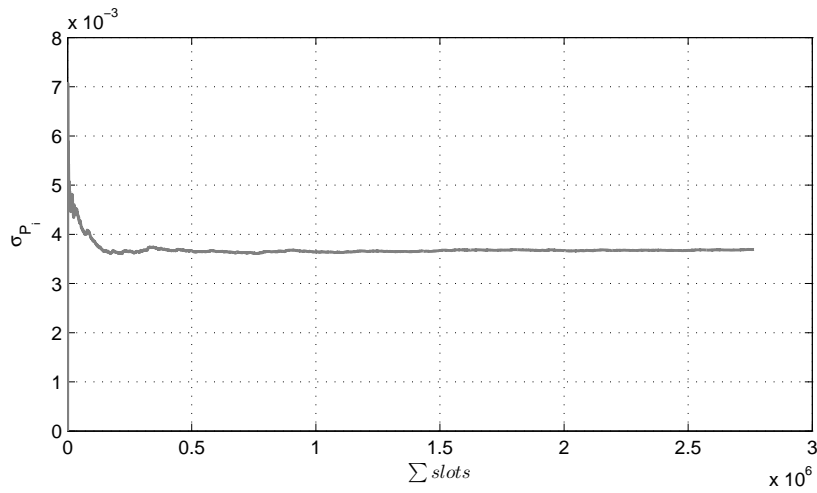


Figura 4.11: Variação do \bar{P}_{idle} com o tempo.

Observe-se que na Figura 4.12 o desvio padrão de P_i inicia com variações até chegar ao ponto em que se mantém constante, que sucede após ter passado os 10 segundos (equivalente a 50000 *slots*).

Para o efeito da optimização da janela de contenção (Cw), foi desenvolvido um algoritmo inspirado no protocolo *Idle Sense* apresentado em [17], que faz a leitura do P_i no buffer, e com base no valor lido altera a janela, de forma a ter o P_i sempre o mais próximo de P_i^* (ver Algoritmo 4.1). Com algumas medições ao longo do desenvolvimento deste trabalho, decidiu-se adoptar os valores para α e ϵ de 1.001 e 10^{-6} , respectivamente.

Na Figura 4.13 é ilustrado o diagrama de blocos que traduz o algoritmo 4.1, onde a

Figura 4.12: Desvio padrão do P_i .

componente de saída (alteração da janela de contenção(Cw)) é baseada na comparação entre a componente de entrada (P_i) com o parâmetro P_i^* .

Durante o estado de contenção os nós calculam o P_i visto pelo meio;

parâmetros de entrada: P_i

parâmetros de Saída : Cw

if ($P_i \geq P_i^*$) **then**

 Aumentar o $\tau \Rightarrow$ Diminuir o Cw ;

$Cw \leftarrow \frac{2}{2+\epsilon Cw}$;

else

 Diminuir o $\tau \Rightarrow$ Aumentar o Cw ;

$Cw \leftarrow \frac{Cw}{\alpha}$;

end

Algoritmo 4.1: Algoritmo para otimização da janela

Figura 4.13: Diagrama de blocos do algoritmo.

Capítulo 5

ANÁLISE DO DESEMPENHO

Este capítulo apresenta a análise do desempenho do protocolo implementado e descrito no capítulo anterior. Para uma melhor rectidão na análise, comparam-se os resultados com o protocolo 802.11. As métricas utilizadas na comparação dos protocolos baseiam-se essencialmente no débito (número de tramas enviadas com sucesso), bem como a taxa sucesso.

Este capítulo encontra-se dividido em quatro secções. Na primeira secção, uma vez que o protocolo desta dissertação foi implementado recorrendo ao simulador ns-2, é apresentada uma breve descrição sobre o simulador. Na segunda secção descrevem-se os vários parâmetros utilizados na simulação. Na terceira secção é descrito o método de recolha de dados das simulações, bem como o cálculo de débito e probabilidade de sucesso. Por fim, são apresentados os resultados obtidos nos testes realizados.

5.1 O simulador

O simulador usado foi o “*The Network Simulator ns-2*” [9], que é um simulador *open source* baseado em eventos discretos e especialmente destinado para o âmbito da investigação científica. O simulador cobre uma vasta gama de aplicações, de protocolos, de tipos de redes, de elementos da rede e modelos de tráfego. São adoptadas duas linguagens de programação: uma de script OTcl [18], e outra de programação C++. A implementação de baixo nível é totalmente baseado em C++ e os scripts tcl são usados como comandos e interfaces de configuração [19].

Uma simulação típica do ns-2 produz ficheiros trace contendo dados dos pacotes de *routing* e pacotes de dados, localização corrente dos nós, entre outros, que depois podem ser analisados para obter dados estatísticos (ver Figura 5.1).

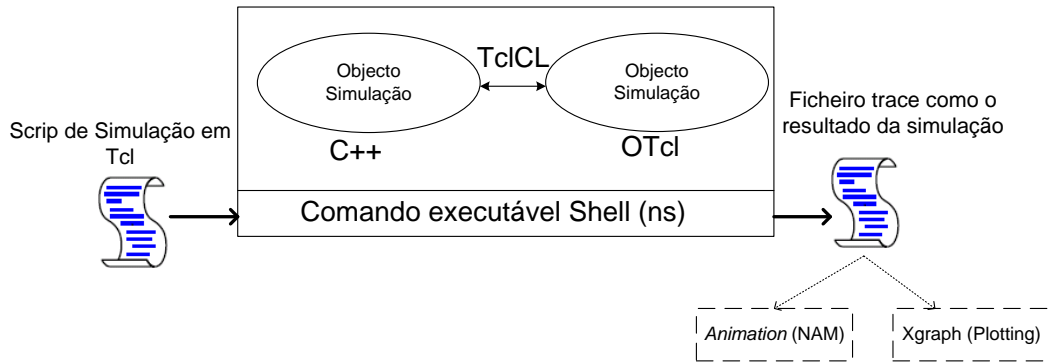


Figura 5.1: Arquitectura ns2 em que o OTcl representa a interface do utilizador, C++ o mecanismo interno e o Tclcl interliga C++ ao OTcl.

O ns-2 foi criado para redes com fios. Como o resultado do projecto *Monarch* desenvolvido na CMU (*Carnegie Mellon University*), um grupo de investigação implementou a funcionalidade *wireless* no ns. Alguns protocolos de *routing* em redes ad hoc e a funcionalidade da camada física de 802.11 foram também implementadas [20].

5.2 Parâmetros Usados no Simulador

A Tabela 5.1 resume os parâmetros usados no simulador, e nas próximas subsecções é explicado em detalhe as opções tomadas ao longo do desenvolvimento. No anexo B é apresentado o script TCL usado nas simulações.

SIFS	10 μs	ritmo de transmissão de dados	11 Mbps
DIFS	50 μs	ϕ	416 μs
EIFS	364 μs	ACK	304 μs
σ	20 μs	Atraso de Propagação (δ)	1 μs
W_0	1000	Tempo de Simulação	200 s
Comprimento da fila de espera (K)	50	Comprimento das tramas	1500 bytes
Comprimento da janela de amostragem (B)	500	Mecanismo de transmissão	básico

Tabela 5.1: Parâmetros utilizado na simulação.

Ao longo da elaboração deste trabalho, foi considerado que os nós permanecem imóveis nas suas posições iniciais durante toda a simulação.

Tamanho das tramas de dados

Neste trabalho caracteriza-se o desempenho quando a rede se encontra totalmente saturada, ou seja, com a fila de espera de transmissão dos nós completamente cheia. O tamanho das tramas foi considerado fixo e de 1500 bytes. Contudo, o que realmente define a carga da rede são o composto do tamanho das tramas com o ritmo a que são enviadas as tramas, e ainda o número dos nós que geram as tramas.

Ritmo de geração das tramas

Em toda a simulação os nós geram tramas a um ritmo constante com um período de geração de $1.5ms$, o que totaliza cerca 666 [pacotes/segundo/nó], sendo o ritmo de transmissão tanto para tráfego *unicast* como *broadcast* de 11Mbps.

Geração do tráfego

O simulador já é distribuído com geradores clássicos de tráfego implementados, tais como aplicações CBR, TCP, Sink, FTP, etc. No entanto, como o estudo é específico para a subcamada MAC, foi usado um gerador de tráfego desenvolvido na equipa de investigação, de forma a que os nós escolham o endereço de destino consoante o tipo de tráfego (*unicast* ou *broadcast*) e agendando a transmissão. Este gerador de tráfego regista toda actividade da fila de espera de transmissão (secção 5.3).

5.3 Aquisição de Dados e Cálculo do Débito

Para aquisição dos dados foi criado um ficheiro (*Logger*) em que se regista toda a actividade dos nós durante a simulação. O Logger contém 7 itens:

1. Identificador do nó;
2. Identificador da mensagem;
3. Tempo em que a mensagem foi colocada na fila de espera;
4. Tempo em que a mensagem foi retirada da fila de espera;
5. Instante da recepção da mensagem no receptor;

6. Tipo de transmissão (*Broadcast/Unicast*);
7. Tamanho da fila de espera no instante em que a trama é inserida na fila.

O identificador do nó (item 1) apresenta o número dos nós da topologia vista na secção 4.1 (0 a 99 nós). O segundo item diz respeito à sequência das tramas. O sexto item regista o valor 1 quando se trata de transmissão *Broadcast* e 0 para transmissão *unicast*. O sétimo e último item indica o estado/tamanho da fila de espera. Quando estiver a -1 é porque houve um drop na fila de espera, e a 49 indica a fila cheia. Os restantes itens servem para o efeito assim como os nomes apresentam. Um exemplo típico do trace no ficheiro é

```
6  313  0.180398  0.697963  0.744627  0  49
```

que descreve que trama com id 313, foi recebida no nó 6, no instante 0.180398s e foi retirada da fila de espera no instante 0.697963s. Além disso, quando a trama é retirada da fila, esta é recebida com sucesso por um outro receptor no instante 0.744627s. O valor 0 do sexto item indica que a transmissão foi do tipo *broadcast* e por último o valor 49 indica o número de tramas contidas na fila de espera de transmissão.

O débito é definido como o rácio entre todas as tramas transmitidas (item 4) sobre o tempo total da simulação enquanto que o débito útil considera somente tramas enviadas e que são recebidas no nó de destino (item 5) sobre a duração da simulação.

5.4 Desempenho

De forma a analisar o desempenho do protocolo implementado, foi efectuada a comparação deste com o protocolo da norma IEEE 802.11 numa rede com a topologia da secção 4.1 e considerando os parâmetros já mencionados anteriormente na Tabela 5.1. Na figura 5.2, é comparado o débito obtido pelos dois protocolos. Observa-se que o protocolo 802.11 atinge débitos elevados em cenários com presença de transmissões *Unicast*. Porém para cenários puramente *Broadcast* (100% *broadcast*) já não acontece o mesmo, isso porque na transmissão *broadcast* o protocolo utiliza uma janela fixa ($Cw=32$) que não é adaptada ao número dos nós no meio. Por outro lado, o protocolo proposto (Capítulo 4) apresenta um comportamento praticamente constante tanto na presença de transmissões *unicast*, como *broadcast* puro, porque a janela de contenção é sempre adaptável à actividade do

meio. Este comportamento do protocolo proposto na presença de tráfego exclusivamente *broadcast* faz com que seja bastante útil para o uso nos protocolos de *routing* em MANET (*Mobile Ad Hoc Networks*). A Figura 5.3 representa a probabilidade de sucesso da

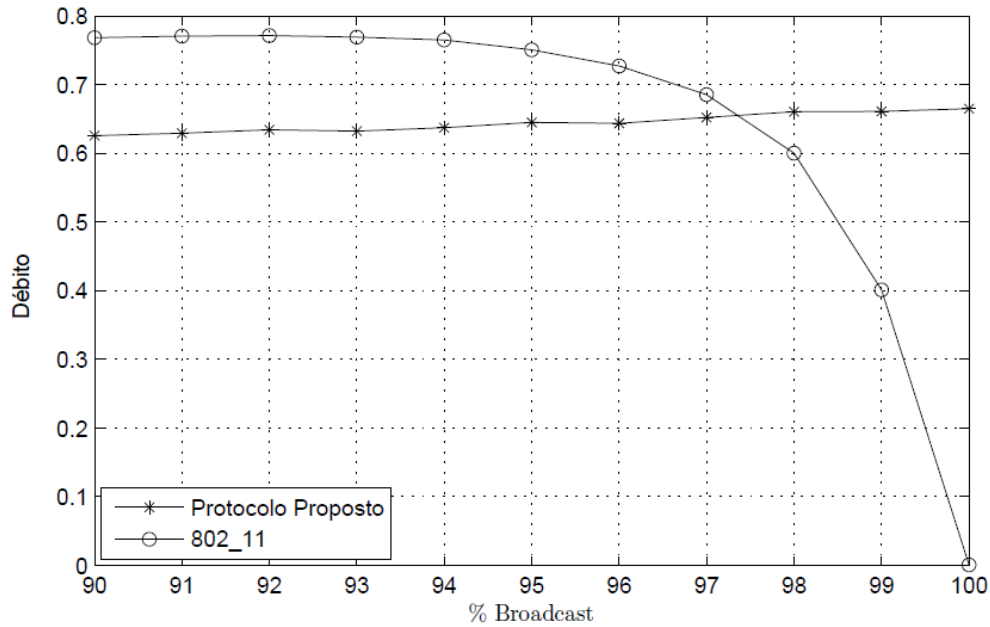


Figura 5.2: Débito dos protocolos IEEE 802.11 e do protocolo proposto no Capítulo 4 em função da percentagem de tramas *broadcast* geradas.

transmissão das tramas. Note-se que no protocolo 802.11, o aumento de tramas *broadcast* conduz à diminuição da probabilidade de sucesso, pois quanto menos tramas *unicast* transmite, menor serão as retransmissões. É de salientar que a fiabilidade da transmissão varia directamente com a quantidade de retransmissão das tramas (como acontece no BMW [11]). Já no protocolo proposto, a probabilidade de sucesso mantém-se constante com a variação de tráfego de *broadcast* porque a janela é sempre adaptada ao meio independentemente do tipo de tramas a serem enviadas (*unicast* ou *broadcast*).

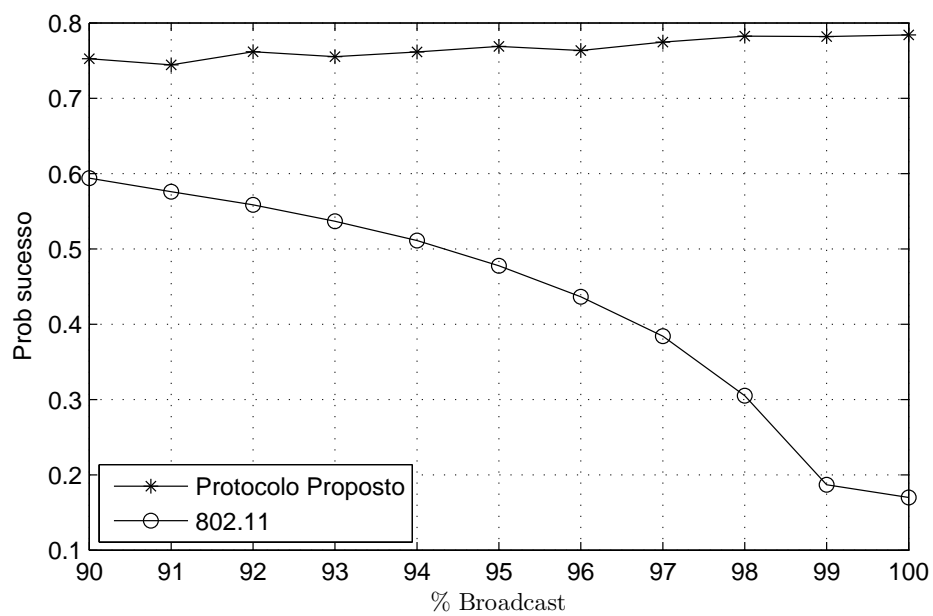


Figura 5.3: Probabilidade de sucesso dos protocolos IEEE 802.11 e do protocolo proposto em função da porcentagem tramas *broadcast* geradas.

Capítulo 6

CONCLUSÕES

Neste último capítulo é realizada uma análise geral sobre o trabalho desenvolvido. De seguida são resumidos os principais contributos desta dissertação.

6.1 Conclusões

Nesta dissertação foram abordados temas sobre uma área considerada desafiante actualmente, que são as redes ad hoc. Em particular foi abordado o controlo de acesso ao meio dos nós de forma distribuída na perspectiva de que há vários nós competindo pelo acesso à largura de banda limitada disponível numa rede sem fios. Daí a necessidade dos nós terem de optimizar as suas janela de contenção de forma a usarem o canal com o máximo proveito possível, diminuindo com isso a probabilidade de ocorrerem colisões no meio e aumentando o débito útil do sistema. Existem dois factores que tornaram o desenvolvimento desta dissertação tão aliciante. O primeiro foi o facto do estudo se incluir na área científica das telecomunicações, presente no dia-a-dia e ainda com mais demanda num futuro próximo. Contudo, também foi motivante o estudo e a implementação de um novo protocolo com uma abordagem totalmente diferente do protocolo que actualmente é usado nas redes de comunicação sem fios (norma IEEE 802.11).

Ao verificar os trabalhos que envolvem a sub-camada MAC da pilha protocolar do modelo OSI, propostos pela comunidade científica, mais concretamente os trabalhos que envolvem comunicação de redes ad hoc, nota-se que alguns possuem virtudes em alguns cenários de aplicação. Contudo, o desempenho já não justifica o seu uso quando aplicados

em cenários com carga saturada ou então na presença de redes multi-hop, devido a algumas questões associadas nomeadamente à existência de nós escondidos.

No Capítulo 3, foi proposto um modelo teórico que caracteriza o cálculo da probabilidade de sucesso num cenário de rede multi-hop, e o débito útil como sendo o rácio entre o tempo em que o canal é usado com sucesso e tempo total de utilização. O modelo é explorado no Capítulo 4, obtendo-se a probabilidade de *idle slots* óptimo P_i^* que foi usado como parâmetro de referência na optimização da janela de contenção.

Nos resultados obtidos através das simulações realizadas no Capítulo 5, foi analisado o desempenho do protocolo proposto em comparação com o protocolo da norma IEEE 802.11. O protocolo proposto apresenta uma vantagem em relação ao protocolo 802.11, nomeadamente no compromisso entre o débito útil do sistema e a probabilidade de sucesso na transmissão das tramas. O protocolo proposto garante a maximização do débito, e verifica-se que a fiabilidade é mantida quer para os diferentes tipos de tráfego (*unicast/broadcast*), quer para diferentes densidades de nós.

Apêndice A

Anexo A

Algoritmo da Optimização da Janela de Contenção

Apresenta-se de seguida a implementação do código em C++ integrado no ns2 do algoritmo que efectua a optimização da janela de contenção com base no P_i observado durante o período de contenção. O código consiste basicamente na construção de um *buffer* circular, que quando estiver preenchido, começa actuar na janela consoante a comparação do P_i com o P_i^* .

```
int size_vect_prob_idle = 100; // tamanho do buffer
/* Buffer deslizante*/
if (countt > size_vect_prob_idle-1)
{
    mac->log_stats->list_nodes[mac->index_].acc_idle_slots-=idle_vect[countt % ↵
        size_vect_prob_idle];
    mac->log_stats->list_nodes[mac->index_].count_idle_slots-=sum_vect[countt % ↵
        size_vect_prob_idle];
    idle_vect[countt % size_vect_prob_idle] = (int)(samples-slots_bsy);
    sum_vect[countt %size_vect_prob_idle]   =(int)(samples);

    mac->log_stats->list_nodes[mac->index_].acc_idle_slots+=idle_vect[countt % ↵
        size_vect_prob_idle];
    mac->log_stats->list_nodes[mac->index_].count_idle_slots+=sum_vect[countt % ↵
        size_vect_prob_idle];
    countt ++;

    if(mac->log_stats->list_nodes[mac->index_].count_idle_slots != 0)
```

```

Pi =(float)(mac->log_stats->list_nodes[mac->index_].acc_idle_slots/mac->log_stats->list_nodes[mac->index_].count_idle_slots);

/* Ajustamento da janela de contenção* Ref Pi=0.92 */
if(Pi < 0.992 )
{
    mac->cw_ = (int) (mac->cw_ *1.001);
}
else
{
    mac->cw_ = (int) (2.0*mac->cw_)/(2.0+mac->cw_ * pow(10,-6));
}

//Limitar o Cw
if (mac->cw_ <= 200)
    mac->cw_ =200;
else if (mac->cw_ > 40000)
    mac->cw_ = 40000;
}

/* Preenchimento do Buffer*/
else
{
    idle_vect[countt] = (int)(samples-slots_bsy);
    sum_vect[countt] =(int)(samples);
    countt++;
    mac->log_stats->list_nodes[mac->index_].acc_idle_slots +=(int)(samples-slots_bsy);
    mac->log_stats->list_nodes[mac->index_].count_idle_slots+=(int)(samples);
}

```

Apêndice B

Anexo B

Script Tcl de simulação em ns2

```
%\begin{tcl}
remove-all-packet-headers
add-packet-header LDP Pushback Ping ARP MAC_GEN_STATS LL Mac NetIF IFq IP Common ←
    Flags
#=====
#                               Variaveis
#=====
set opt(x)      990
set opt(y)      990
set opt(nn)     100
set opt(stop)   200
set opt(tr)     out_802.tr
set opt(trnam)  out_802_11.nam
set val(tp)     1;
set val(dens)   1;
set val(grelha) "$"topo_n100_d90.tcl"$";
#=====
#                               MAC
#=====
Mac/802_11 set dataRate_      11.0e6
Mac/802_11 set basicRate_     11.0e6
Mac/802_11 set RTSThreshold_  3000

##### Phy #####
Phy/WirelessPhy set RXThresh_ 1.42681e-08; #100 meters
```

```
Phy/WirelessPhy set CStresh_ 0.92681e-08; #111.4 meters
```

```
=====
```

```
# Configuração dos nós
```

```
=====
```

```
set opt(seed) random seed;
if {$opt(seed) > 0} {
    ns-random $opt(seed)
}
```

```
set ns_ [new Simulator]
set tracefd [open $opt(tr) w]
$ns_ trace-all $tracefd
```

```
set topo [new Topography]
$topo load_flatgrid $opt(x) $opt(y)
create-god $opt(nn)
```

```
$ns_ node-config -adhocRouting DumbAgent \
    -llType LL \
    -macType Mac/802_11\
    -ifqType Queue/DropTail \
    -ifqLen 50 \
    -antType Antenna/OmniAntenna \
    -propType Propagation/TwoRayGround \
    -phyType Phy/WirelessPhy \
    -channelType Channel/WirelessChannel \
    -topoInstance $topo \
    -agentTrace OFF \
    -routerTrace OFF \
    -macTrace OFF \
    -movementTrace OFF
```

```
for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node ]
    $node_($i) random-motion 0 ;# disable random motion
}
```

```
source $val(grelha); # topologia da grelha com 100 nós
```

```
=====
```

```
# Agents
```

```
=====
```

```
for {set i 0} {$i < $opt(nn)} {incr i 1} {
```

```

    set p($i) [new Agent/MAC_GEN_STATS $i]
    # associa o agent MAC_GENT_STATS ao nó
    $ns_ attach-agent $node_($i) $p($i)
    # associa o agent ao porto 290, porto do agent MAC_GENT_STATS
    $node_($i) attach $p($i) 290
}

for {set i 0} {$i < $opt(nn)} {incr i} {
    set mac_($i) [$node_($i) getMac 0];
    ## manter esta ordem de atribuição de parametrização
    $ns_ at 0.0000001 "$p($i) set_num_nodes $opt(nn)";
    $ns_ at 0.0000001 "$p($i) set_percent_broadcast 100.0000";
    $ns_ at 0.0000001 "$p($i) set_mode 0";
    $ns_ at 0.0000001 "$p($i) set_lambda 666";
    $ns_ at 0.0000001 "$p($i) topology $val(tp)";
    $ns_ at 0.0000001 "$p($i) density $val(dens)"; # número de nós em cada posição
    $ns_ at 0.0000000001 "$mac_($i) set_num_nodes $opt(nn)";
    $ns_ at 0.000000011 "$mac_($i) create_queue_logger";
    ## Começa o envio em cada agente
    $ns_ at 0.00000111 "$p($i) start";
    $ns_ at $opt(stop) "$p($i) print";
}

#=====
#                               Fim
#=====

Agent/Ping instproc recv {from rtt} {
    $self instvar node_
    puts "node [$node_ id] received ping answer from \
        $from with round-trip-time $rtt ms."
}

$ns_ at $opt(stop).9 "$ns_ halt"
puts "Starting Simulation..."
$ns_ run
% \end{tcl}

```


Bibliografia

- [1] S.K.S. Gupta, V. Shankar, and S. Lalwani. Reliable Multicast MAC Protocol for Wireless LANs. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 1, pages 93 – 97 vol.1, may 2003.
- [2] M. Luis, R. Oliveira, L. Bernardo, and R. Dinis. A Reliable Broadcast and Unicast MAC Protocol for Ad Hoc Networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1 –6, may 2011.
- [3] Kevin J. Negus, John Waters, Jean Tourrilhes, Chris Romans, Jim Lansford, and Stephen Hui. HomeRF and SWAP: Wireless Networking for the Connected Home. *SIGMOBILE Mob. Comput. Commun. Rev.*, 2(4):28–37, October 1998.
- [4] J. Redi and B. Welsh. Energy-conservation for Tactical Robot Networks. In *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, volume 2, pages 1429 –1433 vol.2, 1999.
- [5] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Commun. ACM*, 43(5):51–58, May 2000.
- [6] J. Jubin and J.D. Tornow. The DARPA Packet Radio Network Protocols. *Proceedings of the IEEE*, 75(1):21 – 32, jan. 1987.
- [7] N. Shacham and J. Westcott. Future Directions in Packet Radio Architectures and Protocols. *Proceedings of the IEEE*, 75(1):83 – 99, jan. 1987.
- [8] Grupo do trabalho IETF MANET. Ver <http://www.ietf.org/html.charters/manet-charter.html>.

- [9] Information Sciences Institute. NS-2 network simulator (version 2.31). Software Package retrieved from <http://www.isi.edu/nsnam/ns/>, 2007.
- [10] IEEE. IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std 802.11, 1999 Edition (R2003)*, pages i –513, 2003.
- [11] K. Tang and M. Gerla. MAC Reliable Broadcast in Ad Hoc Networks. In *Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE*, volume 2, pages 1008 – 1013 vol.2, 2001.
- [12] Min-Te Sun, Lifei Huang, A. Arora, and Ten-Hwang Lai. Reliable MAC Layer Multicast in IEEE 802.11 Wireless Networks. In *Parallel Processing, 2002. Proceedings. International Conference on*, pages 527 – 536, aug. 2002.
- [13] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '99*, pages 151–162, New York, NY, USA, 1999. ACM.
- [14] Shiann-Tsong Sheu, Yihjia Tsai, and Jenhui Chen. A Highly Rreliable Broadcast Scheme for IEEE 802.11 Multi-hop Ad Hoc Networks. In *Communications, 2002. ICC 2002. IEEE International Conference on*, volume 1, pages 610 –615, 2002.
- [15] J. Kuri and S.K. Kasera. Reliable Multicast in Multi-access Wireless LANs. In *IN-FOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 760–767 vol.2, Mar 1999.
- [16] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535 –547, march 2000.

- [17] Martin Heusse, Franck Rousseau, Romaric Guillier, and Andrzej Duda. Idle Sense: An Optimal Access Method for High Throughput and Fairness in Rate Diverse Wireless LANs. *SIGCOMM Comput. Commun. Rev.*, 35(4):121–132, Aug. 2005.
- [18] Tcl Developer Xchange. Ver <http://www.tcl.tk>, 2007.
- [19] Kevin Fall and Kannan Varadhan. ns Notes and Documentation, 2000.
- [20] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '98, pages 85–97, New York, NY, USA, 1998. ACM.

